

NASA/CR-2015-218774



# An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Sixth Revision

*Terence S. Abbott  
Stinger Ghaffarian Technologies, Hampton, Virginia*

## NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

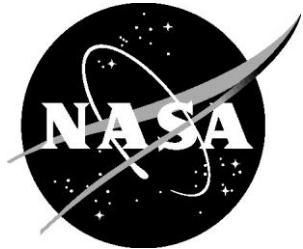
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:  
NASA STI Information Desk  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

NASA/CR-2015-218774



# An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Sixth Revision

*Terence S. Abbott  
Stinger Ghaffarian Technologies, Hampton, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

Prepared for Langley Research Center  
under Contract NNL10AA14B

---

July 2015

Available from:

NASA STI Program / Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199  
Fax: 757-864-6500

## Table of Contents

Definitions.....	iii
Acronyms and Nomenclature.....	iv
Introduction.....	1
Overview.....	2
Trajectory-Based Operation Algorithm Implementation .....	5
Trajectory Computation .....	6
General.....	6
Final Approach Speed .....	7
Initial Cruise Altitude and Speed .....	7
Top of Descent Monitoring.....	8
Recalculation of the Vertical Path.....	8
Wind Forecast Data.....	8
Trajectory State Data Computation.....	10
Calculation of the Spacing Interval.....	12
Basic TBO Time Interval .....	12
Basic TBO Distance Interval.....	12
Achieve-By and Termination Point at the Final Approach Fix.....	12
TBO Speed Control Law.....	13
TBO Speed Change Minimization and Lag Compensation .....	16
TBO End Speed Estimation .....	17
TBO Speed Command Quantization.....	18
TBO Nominal Deceleration Roll-In Logic.....	18
Look Ahead Speed Change Inhibit .....	19
Stated-Based, Constant-Time Delay Algorithm .....	20
CTD Design Considerations and General Implementation Notes.....	22
Ownship state and TBO data.....	24
FIM Clearance.....	24
ALL Traffic State Data .....	24
Process and Record Traffic Data.....	25
Compute the Nominal Speed .....	25
Compute the Spacing Error.....	25
Compute and Limit the Speed Error .....	25
Determine if a Speed Change is Required .....	26

Calculate the Speed Change.....	27
Apply Profile and Configuration Speed Limits .....	30
Limits based on the nominal profile speeds .....	30
Airframe limits .....	31
Procedural limits .....	31
Linearize Intermediate Speed Command .....	31
Condition 1.....	31
Condition 2.....	32
Apply Procedural 10,000 ft / 250 kt Speed Limit .....	33
Switch to Final Approach Speed.....	33
Distance-Based Clearance.....	33
Control Law Selection .....	34
Operational Considerations.....	36
Common Speeds After Merging .....	36
Envelope Protection .....	36
Off-Nominal Mach / CAS Transition .....	36
Future Design Considerations.....	36
Estimated Time of Arrival Data from the Traffic to Follow.....	36
Trajectory Data from the Ownship FMS .....	37
Speed Prediction, Speed Quantization, and Lag Compensation Functions .....	37
Summary .....	37
References.....	39

## Definitions

Achieve-by point:	A designated waypoint where the assigned spacing goal is expected to be achieved. It is also the waypoint where the spacing technique should transition from trajectory-based to state-based.
Assigned spacing goal:	The spacing interval, either in time or distance, assigned by ATC in the spacing clearance.
Constant time delay:	A specific state-based spacing algorithm that uses the measured spacing interval (MSI) to define the spacing error.
End speed:	The projected or expected speed command at the end of a change in the commanded speed. I.e., this is the expected, steady-state speed command at the end of a speed change.
End speed command:	See End speed.
Federated system:	An aircraft system that is designed as a stand-alone implementation, i.e., not integrated into the existing avionics.
FIM aircraft or ownship:	The aircraft that is using its FIM equipment, e.g., an ASTAR13 implementation, to conduct a self-spacing operation.
Measure spacing interval:	Using the stored state data (time, position, ground speed, and ground track angle) from the lead aircraft, the measured spacing interval in time is the difference between the current time and the time when the lead aircraft was at the FIM aircraft's proximate position. The measured spacing interval in distance is the along-path distance between FIM aircraft and the lead aircraft.
Ownship:	See FIM aircraft.
Planned termination point:	A designated waypoint where the FIM operation is to terminate. For distance based clearances, the FIM operation terminates when the lead aircraft passes this point. For time based clearances, the FIM operation terminates when the FIM aircraft passes this point.
Spacing error:	The error value that the control law is attempting to minimize. The trajectory-based and state-based algorithms use different methods for calculating the spacing error. Additionally, the spacing error may be defined in either time or distance, depending on the assigned spacing goal.
Speed command:	The continuous, instantaneous speed command provided by the algorithm.
State-based operations:	Based on the previous and current physical states (time, position, ground speed, and ground track angle) of the lead and FIM aircraft. These operations are only valid when the FIM and lead aircraft are on a common path.

Time-to-go:	The time to go from the aircraft's current position to a designated point using the TBO prediction calculation.
Traffic to follow:	The aircraft against which the spacing aircraft is performing a spacing operation. I.e., the lead aircraft.
Trajectory-based operations:	Operations based on a calculated 4-D path, typically from the aircraft's current position to the runway threshold. The calculated 4-D paths are used to calculate the time-to-go of the FIM and lead aircraft. The difference in the estimated time-to-go of the FIM and lead aircraft to a common point is used to define the spacing error.

## **Acronyms and Nomenclature**

2D:	2 dimensional; longitudinal and lateral
4D:	4 dimensional; longitudinal, lateral, vertical, and temporal
ABP:	Achieve-by point
ADS-B:	Automatic Dependence Surveillance Broadcast
ASG:	Assigned spacing goal
ASTAR:	Airborne Spacing for Terminal Arrival Routes
ATC:	Air traffic control
ATD-1:	Air Traffic Management Technology Demonstration-1
CAS:	Calibrated airspeed
CTD:	Constant time delay
DTG:	Distance-to-go
ETA:	Estimated time of arrival
FAF:	Final approach fix
FIM:	Flight-deck interval management.
FMS:	Flight management system
ft:	Foot/feet
gs:	Ground speed

IM:	Interval management
kt:	Knots
MSI:	Measure spacing interval
nmi:	Nautical miles
PTP:	Planned termination point
RTA:	Required time of arrival
SE:	Spacing error
STAR:	Standard Terminal Arrival Route
TBO:	Trajectory-based operations
TCP:	Trajectory change point
TOD:	Top-of-descent
TTF:	Traffic to follow
TTG:	Time-to-go

## Abstract

*This paper presents an overview of the sixth revision to an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This algorithm is referred to as the Airborne Spacing for Terminal Arrival Routes version 13 (ASTAR13). This airborne self-spacing concept contains both trajectory-based and state-based mechanisms for calculating the speeds required to achieve or maintain a precise spacing interval. The trajectory-based capability allows for spacing operations prior to the aircraft being on a common path. This algorithm was also designed specifically to support a standalone, non-integrated implementation in the spacing aircraft. This current revision to the algorithm adds the state-based capability in support of evolving industry standards relating to airborne self-spacing.*

## Introduction

Concepts for self-spacing of aircraft operating in an airport terminal area have been under development by the National Aeronautics and Space Administration (NASA) since the 1970's (ref. 1). Interest in these concepts have recently been renewed due to a combination of emerging, enabling technology (Automatic Dependent Surveillance Broadcast data link, ADS-B) and the continued growth in air traffic with the ever increasing demand on airport and runway throughput. Terminal area self-spacing has the potential to provide an increase in runway capacity through an increase in the accuracy of over-the-threshold runway crossing times (ref. 2).

A follow-on to NASA's terminal area in-trail spacing development (refs. 3 and 4) and the initial development of a concept and implementation of a trajectory-based merging capability (ref. 5) was instantiated in an application called the Airborne Spacing for Terminal Arrival Routes, ASTAR. This concept extended the self-spacing capability beyond the terminal area to a point prior to the top of the en route descent. This implementation was a trajectory based concept for the entire arrival spacing operation. The second revised implementation of this algorithm (ref. 6) was designed to support dependent runway operations and was referred to as ASTAR10. This second revision provided the ability to manage spacing against two traffic aircraft, with one of these aircraft operating to a parallel runway. This support for parallel dependent runway operations also included the computation of offset threshold crossing times based on the longitudinal distance offset between the two parallel runways and the ability to use diagonal distance spacing once the aircraft are on parallel approaches (ref. 7). This revision of ASTAR also had a rewritten control law relative to the previous version that was based on the original Advanced Terminal Area Approach Spacing (ATAAS) algorithm (ref. 3). The third revision of ASTAR, referred to as ASTAR11 (ref. 8), was a "lite" version of ASTAR10. In this revision, the ability to space against a second aircraft that is operating to a dependent runway was removed. Additionally, several implementation improvements were made based on observations from a pilot-in-the-loop simulation using ASTAR10. The fourth revision of ASTAR (ref. 9) was another update to ASTAR11. In this revision, the primary focus was to modify ASTAR to better support the flight deck interval management portion of the Air Traffic Management Technology Demonstration-1 (ATD-1) (ref. 10). Part of this modification included a change to the basic control law and the ability to recalculate an aircraft's vertical path if that aircraft is off of its preplanned vertical path. The fifth revision of ASTAR (ref. 11), referred to as ASTAR12, was an update to ASTAR11. In this revision, the focus was to again modify ASTAR to better support the flight deck interval management portion of ATD-1, where the ATD-1 environment does not support the rich data exchange envisioned for the NextGen environment (ref. 12). Due to this data sharing limitation between the ground

systems and the aircraft, a ground speed compensation term was added to the control law to account for some of the operational differences between the scheduled times of arrival used by the ATD-1 ground tools and this airborne tool, where these differences are not communicated to the aircraft due to the voice-only ATC environment in ATD-1. For example, the ground tools typically include a schedule delay during high demand periods at an airport. Without knowledge of this schedule delay, the airborne system may issue speed commands that seem inappropriate to ATC. Additionally, a ground speed feedback term could eliminate some of the spacing error at the final approach fix (FAF) during situations where the traffic aircraft is flying a constant speed offset from the planned, nominal speed.

In addition to the trajectory-based operation (TBO) concepts like ASTAR12, earlier work also examined the use of state-based concepts for Flight-deck Interval Management (FIM), i.e., airborne self-spacing. These state-based concepts do not require any knowledge of the leading aircraft's planned route (refs. 13-16) and typically use saved aircraft state data recorded from the leading aircraft. While these state-based concepts could easily support self-spacing operations, they could not provide the potential benefits in airport noise reduction and reduced fuel usage that are offered by TBO concepts. However, because these state-based concepts do not require knowledge of the leading aircraft's intended flight path, they are not susceptible to path prediction errors encountered in the TBO concepts.

To define FIM equipment requirements for near-term operational environments, the RTCA has developed a document that describes equipment requirements for FIM operations. Partly because the near-term ATC environment does not support the data exchange necessary for an optimized TBO-based FIM operation, the RTCA has defined several FIM capabilities and requirements that are better suited for this near-term period. The primary FIM capabilities defined by RTCA require the use of an integrated TBO and state-based concept for most FIM operations. The focus of this document is the description of the development and integration of a new state-based spacing capability into ASTAR's existing TBO concept.

This sixth revision of ASTAR, referred to as ASTAR13, is a major change to ASTAR. In this revision, the primary focus was to further modify ASTAR to better support the flight deck interval management portion of ATD-1, with the ATD-1 FIM requirements based on the evolving industry requirements described in the draft RTCA document titled *Minimum Operational Performance Standards (MOPS) for Flight-deck Interval Management (FIM)* (ref. 17), otherwise known as the FIM MOPS. As a requirement of this document to support several new FIM clearances, a separate, state-based speed control law was integrated into the ASTAR framework, with additional logic to determine which of the two control laws to employ. This dual control law concept, ASTAR13, is described in the subsequent text.

## Overview

The RTCA FIM MOPS describes five FIM clearance types: Maintain Current Spacing, Capture Then Maintain Spacing, Achieve-by Then Maintain, Final Approach Spacing, and IM Turn. Prior versions of ASTAR only supported the Achieve-by portion of the Achieve-by Then Maintain clearance and did not support the other FIM clearances. This new implementation, ASTAR13, is designed to support all of these clearances except for IM Turn. In all of the supported clearances, ASTAR ceases providing speed commands for spacing at the planned termination point (PTP), which is a point on the ownship's route that is either designated by ATC or is part of the planned procedure. For the Achieve-by Then Maintain clearance, ASTAR uses a TBO solution until the designated achieve-by point (ABP) is reached, and a state-based solution is used afterward. A simple description of each of the supported clearances is as follows:

**Maintain Current Spacing Clearance:** The Maintain Current Spacing clearance is intended to be used when ATC desires the FIM aircraft to maintain its current relative spacing, either in time or distance, from its designated lead aircraft. The aircraft must be on a common path for this operation. At the initiation of this clearance, the measured spacing interval (MSI) at the time of initiation becomes the assigned spacing goal (ASG). In this operation, the state-based algorithm is used as the control law.

*Capture then Maintain Spacing Clearance:* From an algorithm standpoint, the Capture then Maintain Spacing Clearance is similar to the Maintain Current Spacing clearance with the exceptions that the ASG is specifically assigned by ATC and that the spacing error is initially allowed to be outside of the acceptable spacing tolerance, i.e., the capture phase. The aircraft must be on a common path for this operation.

*Achieve-by Then Maintain Clearance:* A generic scenario for the Achieve-by Then Maintain Clearance would begin with the FIM and lead aircraft on different routes. ATC would issue the FIM clearance and spacing would begin using the TBO algorithm. Once the FIM aircraft passes the achieve-by point (ABP), a waypoint that must be at or after a point where both aircraft are on a common route, ASTAR would transition into a Maintain Spacing mode using the state-based algorithm.

*Final Approach Spacing Clearance:* The Final Approach Spacing Clearance is similar to the Achieve-by Then Maintain Clearance except that it is only issued when one or both of the aircraft are on the final approach course and the achieve-by point is always the planned termination point. This latter requirement dictates that only the TBO algorithm within ASTAR13 is used.

From a design standpoint, ASTAR13 can conceptually be thought of as two distinct self-spacing speed control laws with logic to determine which control law is to be invoked. The first control law is the TBO concept of ASTAR12. The second control law is a new state-based concept based on a design described in reference 18, which is a constant-time delay (CTD) implementation. The fundamental design of ASTAR13 is shown in figure 1.

A brief summary of the elements shown in figure 1 is provided as follows, with detailed descriptions in the subsequent sections:

- All traffic state data: Input data that contains the ADS-B state data from all of the surrounding aircraft. This would obviously include the state data from the traffic to follow (TTF), if one were selected and its data were available.
- Ownship state and route data: Input data that contains the ownship's inertial and air references data, planned performance data, and augmented route information used in the generation of its 4-D path.
- FIM instruction or clearance data: Input data that contains ASTAR control options and the FIM clearance information. If only the calculation of the MSI is desired, then only the identification of the TTF, i.e., the leading aircraft, is required.
- TTF state and route data: Input data that contains the leading aircraft's inertial data, planned performance data, and augmented route information used in the generation of its 4-D path.
- Process and record all traffic data: Stores approximately 10 minutes of 1 Hz, ADS-B state data from all of the surrounding aircraft.
- Calculate the MSI: If a lead aircraft has been identified and both aircraft are on a common path, then calculate both the along-path distance and along-path time between the ownship and the lead aircraft using the state-based data.

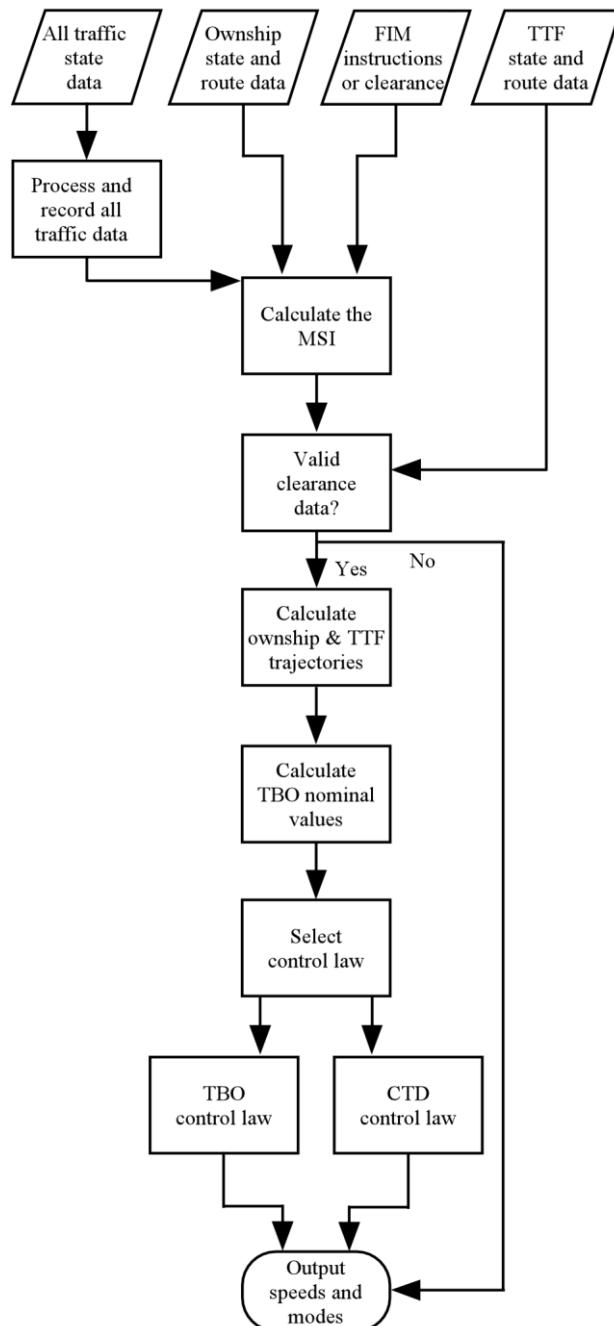


Figure 1. ASTAR13 basic design structure.

- Valid clearance data: As a minimum, valid clearance data would include the FIM clearance type, the identification of the lead aircraft, i.e., the TTF, the identification of the expected routing for the TTF, and the assigned spacing goal. For an Achieve-by Then Maintain Clearance, the identification of the ABP would also be required. The PTP may be identified in the clearance or may be part of the procedure.
- Calculate ownership and TTF trajectories: From the ownership and TTF route information, calculate the planned 4D trajectory for each aircraft.

- Select control law: Depending on the spacing clearance type and the current positions of the aircraft on their respective routes, either the TBO or CTD control law is selected.
- TBO control law: Use the TBO control law to calculate the speed commands.
- CTD control law: Use the CTD control law to calculate the speed commands.
- Output speeds and modes: Output various control mode state data and if appropriate, the speed commands.

## Trajectory-Based Operation Algorithm Implementation

As with the previous versions of ASTAR, the overall concept for a trajectory-based solution for en route and terminal area self-spacing is fairly straightforward. If the 4D trajectory of an aircraft and its position are known, then the aircraft's position on its trajectory can be determined. By knowing the aircraft's position on its trajectory, the aircraft's estimated time-to-go (TTG) to a point is known. To apply this to a self-spacing concept, a TTG is calculated for both the TTF and for the ownship, noting that the trajectories do not need to be the same. The nominal spacing time,  $t_{nominal}$ , and the spacing time error,  $t_{error}$ , can then be calculated as:

$$t_{nominal} = TTG_{TTF} + \text{spacing goal},$$

$$t_{error} = TTG_{ownship} - t_{nominal},$$

where the identification of the TTF aircraft and the determination of the spacing goal is performed by ATC.

A required time of arrival (RTA) capability can also be implemented in a manner similar to the traffic spacing technique. In this case,

$$t_{nominal} = RTA - \text{current time}.$$

From  $t_{error}$ , a speed error value can then be calculated. A conceptual example for the determination of  $t_{error}$  for traffic spacing, i.e.,  $t_{error} = TTG_{ownship} - t_{nominal}$ , is shown in figure 2.

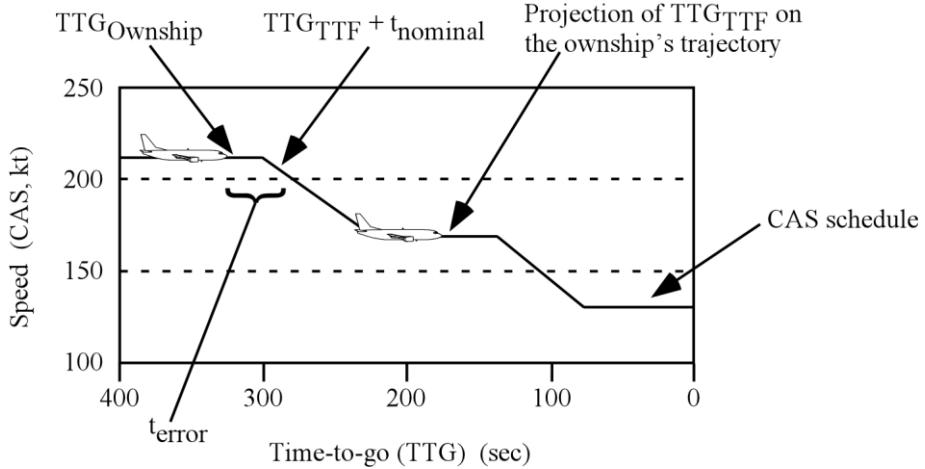


Figure 2. Time error example.

By design, the TBO portion of ASTAR13 is considered an achieve-by algorithm (ref. 19), i.e., it is designed to attain the spacing goal at the achieve-by point, which in the ATD-1 concept is normally the FAF. The algorithm does not exactly obtain and maintain the spacing goal until the ownship is near the achieve-by point. Using this control method, the aircraft should be able to fly speeds that are closer to the nominal profile for a longer portion of the operation relative to a more stringent control method that would maintain a fixed spacing interval.

The implementation of the TBO portion of ASTAR13 is comprised of five major elements: trajectory computation, current trajectory state data computation, the calculation of the spacing interval, the speed control law, and speed change minimization; noting that the trajectory and current trajectory state data computations are also performed when the CTD is used for speed control. Details of these elements are provided in subsequent sections.

## Trajectory Computation

### *General*

The prototype system developed at the NASA Langley Research Center uses a standalone trajectory generator that was developed to calculate a full 4D trajectory from a 2D path specification. Reference 20 provides a description of this algorithm including its input and output parameters. The trajectory definition begins with a simple 2D path definition along with relevant speed and altitude constraints. The 2D path definition is typically an augmented traditional Standard Terminal Arrival Route (STAR) with a continuous connection to an instrument approach procedure. The trajectory generator then computes a full 4D trajectory defined by a series of Trajectory Change Points (TCP's). This standalone approach was developed for two reasons. First, a near-term implementation separate from the flight management system (FMS) was considered to be more practical from a development and implementation cost perspective. Second, since ASTAR needs to calculate the trajectory for the TTF, the additional complexity of calculating the trajectory for the ownship was minimal. Neither of these reasons, however, would preclude use of the FMS for providing the ownship trajectory into ASTAR nor the use of a data linked estimated time of arrival (ETA) or TTG from the traffic aircraft.

One of the major difficulties in computing a 4D trajectory involves the calculation of the length of the ground path during a turn. The turn radius can change due to the presence of wind or changes in the aircraft's specified speed, affecting the length of the ground path. This change in the path length can then affect the distance to a deceleration point, which then affects the turn radius calculation. To accommodate this

interaction, the trajectory calculation uses a multi-pass technique in generating the 4D path with the first pass generating a close approximation to the TCP's based on the computed ground speeds. The following iterations then use the input from the previous pass as a starting point to refine the solution.

In conjunction with the basic 4D calculation, ASTAR preprocesses the trajectory input data. Depending on the situation, ASTAR may update the following generic trajectory parameters: ownship and TTF final approach speeds, initial cruise altitude and speed, differences between the predicted and actual top of descent point, and differences in wind forecast data.

### ***Final Approach Speed***

This option was developed for situations where the achieve-by point was the runway threshold. The use of an achieve-by algorithm coupled with the operational requirement to achieve a stabilized approach means that the algorithm may compensate for differences in the TTF's and the ownship's actual final approach speeds. ASTAR has the ability to modify each aircraft's trajectory data by substituting the individual aircraft's planned final approach speed for the trajectory's generic runway threshold crossing speed. By using the individual aircraft's planned final approach speed, the TTG calculations explicitly compensate for the final approach speed differences between the spacing aircraft. In addition, there are several different operational techniques used to determine where the final approach speed is achieved. In the generic case, the final deceleration starts at a point prior to the runway threshold, causing the aircraft to achieve its final approach speed as it crosses the runway threshold. This baseline technique does not enable stabilized approaches and is not typical for transport aircraft approaches; thus, ASTAR provides two other options. These options are:

- Begin the final deceleration at the waypoint just prior to the runway. This is typical for operations where the final deceleration starts at the final approach fix (FAF).
- Begin a deceleration such that the aircraft reaches its final approach speed at a specific altitude, e.g., to be at the final approach speed at 1000 feet (ft) above the runway's elevation, which is also the minimum requirement for instrument approaches in transport aircraft (ref. 21). To support this option, a special waypoint is included in the trajectory input data and is placed between the runway threshold and the FAF waypoint. Only the crossing altitude and crossing speed data are included in this special waypoint's data and the trajectory generator calculates its position on the horizontal portion of the path.

A fourth option was considered, where the final deceleration starts at a point prior to the FAF such that the aircraft just achieves its final approach speed as it crosses the FAF. This option, however, is not implemented because transport category aircraft do not normally operate in this manner.

### ***Initial Cruise Altitude and Speed***

A second change that ASTAR may make to the ownship's or TTF's trajectory input data is to substitute the individual aircraft's actual cruise altitude and Mach for the initial, generic altitude and Mach specified in the basic augmented 2D path. This change will only occur at the initiation of a new 2D path and with the aircraft's current altitude and Mach matching a relevant data set being provided to ASTAR. That is, the current altitude and Mach of the aircraft must match a special cruise altitude and Mach data set being sent to ASTAR. For the TTF, this special data set could be made available to the ownship via data link.

For the ATD-1 operational environment, neither the cruise speed, the planned decent Mach, nor the planned descent CAS of the TTF are available. Additionally, to further reduce pilot workload, these data are not expected to be available for the FIM aircraft. Because of these data limitations, it was decided to inhibit the generation of speed commands until both the TTF and ownship have passed the first CAS constrained waypoint on their respective routes. In order to calculate viable 4D trajectories, necessary for TBO, reasonably accurate planned speeds are required at critical waypoints along the route. For terminal

airspace operations, the published procedures, general operating rules, and rules specific to that airport typically provide sufficient specificity in defining the majority of the route. However, for the initial cruise and initial descent portion of the operation, speeds can vary greatly based on aircraft type and weight. Because of this, the earlier versions of ASTAR assumed that either the lead aircraft or the ATC system would provide these speeds via capabilities provided in the NextGen environment. Because of this lack of valid cruise and initial descent information in the near-term environment and the detrimental impact on spacing performance and resulting operational acceptability, the 4D trajectory data used in the TBO calculations are not considered to be valid until both aircraft have passed the first waypoint that has a procedural CAS constraint on their respective routes. To support this change, the database containing the basic route information includes the identification of the first published CAS-restricted waypoint on the route.

### ***Top of Descent Monitoring***

The FMS may calculate and the aircraft may fly from a top-of-descent (TOD) point that is appreciably different than the generic TOD estimated by the trajectory generator. Since this difference in the TOD point can introduce a significant error into the estimation of the aircraft's ground speed during this descent and therefore, lead to a significant error in the aircraft's TTG, ASTAR monitors the conformance of the aircraft to its predicted TOD point. If the aircraft begins its descent from cruise before the point that ASTAR predicted, ASTAR will calculate the actual, current descent angle based on this actual TOD and the next altitude crossing restriction, replace the generic descent angle in the augmented 2D path data with this new value, and then recalculate the 4D path. A similar technique is used for a late descent except that ASTAR may recalculate the 4D path several times, depending on how far beyond the originally estimated TOD point that the actual TOD occurs.

### ***Recalculation of the Vertical Path***

Similar to the problem noted in the section *Top of Descent Monitoring*, if an aircraft is significantly far from its planned vertical path, the difference between the planned ground speed and the actual ground speed can be large. This difference in ground speed can then produce a significant error in the aircraft's TTG. Several previous versions of ASTAR allowed the vertical path error, i.e., the difference between the planned altitude along the path and the aircraft's actual altitude at that horizontal position on the path, to reach 6,000 ft, at which time it was assumed that the planned path was no longer valid.

In this version of ASTAR, once an aircraft reaches 4,000 ft of vertical path error, ASTAR will attempt to construct a new 4D path beginning at the aircraft's current estimated position on the original horizontal path. The altitude constraints are then deleted for any downstream waypoint that has an altitude constraint that is higher than the altitude of this new, initial waypoint. Lastly, ASTAR determines if the first altitude constraint following the initial waypoint can be met. If this constraint cannot be met using the original crossing angle, a new crossing angle is calculated based on a linear descent between the initial waypoint and the constraint. This new angle is used in subsequent 4D trajectory calculations.

### ***Wind Forecast Data***

The last modification that ASTAR may apply to the trajectory input data is to modify the original wind forecast data provided to the algorithm. Wind data into and within ASTAR is based on waypoint locations instead of a typical wind grid. It was assumed in the design of ASTAR that a highly developed wind forecast model could be used to provide vertical profile wind data at the waypoint locations. Of special importance to ASTAR would be the wind estimation at the altitude that the trajectory would be crossing the waypoint's position. It was assumed then that the externally provided waypoint wind data would provide reasonably accurate wind data that would bound the expected waypoint trajectory crossing altitude. Up to 10 altitude-wind speed data sets (altitude, direction, and magnitude) per waypoint may be input into ASTAR. From

this initial, external input ASTAR may provide both local and global modifications to the forecast wind data provided to the trajectory generator.

While up to 10 altitude-wind data sets per waypoint may be input into ASTAR, ASTAR itself maintains an internal wind model that uses local aircraft-sensed wind data in addition to the input waypoint wind data. This internal wind model maintains a 1000 ft incremental vertical profile, from 0 ft to 60,000 ft, for every waypoint on all of the paths. This incremental vertical profile contains a "gain" value, the original input wind forecast for this altitude, a measured wind for this altitude, and the current estimated wind forecast for this altitude. Initially, the gain values are all set to zero and the external wind forecast is used to populate the input wind forecast for each altitude in its profile. An altitude-based linear interpolation is used to populate the altitudes that do not directly have any input value.

Measured wind values may be adjusted using local or global data. For the local data case, the ownship's wind derivation is used to update the estimated wind forecast. In this case, wind profiles for every waypoint within 50 nautical miles (nmi) of the ownship's horizontal position may be modified. For these waypoints, if the ownship is at or above 12,000 ft, then each of the 1000 ft incremental vertical profile data sets may be modified for altitudes within  $\pm 5000$  ft of the ownship's altitude. For the situation where the ownship is below 12,000 ft, then each of the 1000 ft incremental vertical profile data sets may be modified for altitudes within  $\pm 3000$  ft of the ownship's altitude. Whether a specific 1000 ft incremental vertical profile data set is modified depends on the current gain value for that data set and the gain value computed for the current ownship's position relative to this data set. The ownship's current gain is calculated as follows:

$x_{ownship}$  = relative horizontal position of ownship (in nmi) to the wind profile point and

$z_{ownship}$  = relative vertical position of ownship (in ft) to the wind profile point.

if  $x_{ownship}$  is greater than 50 nmi,  $gain_{horizontal} = 0$ ,

otherwise  $gain_{horizontal} = 1 - (x_{ownship} / 50 \text{ nmi})$ .

if  $z_{ownship}$  is greater or equal to 12,000 ft, then

if  $z_{ownship}$  is greater than  $\pm 5000$  ft,  $gain_{vertical} = 0$ ,

otherwise  $gain_{vertical} = 1 - \text{absolute value of } (z_{ownship} / 5000 \text{ ft})$ .

otherwise

if  $z_{ownship}$  is greater than  $\pm 3000$  ft,  $gain_{vertical} = 0$ ,

otherwise  $gain_{vertical} = 1 - \text{absolute value of } (z_{ownship} / 3000 \text{ ft})$ .

ownship's current gain =  $gain_{horizontal} * gain_{vertical}$ .

If the ownship's computed gain is greater than the gain value for the data set, then the estimated wind data are updated with the new gain value and measured wind data. The new estimated wind data are computed based on a double linear interpolation between the original forecast winds and the measured winds. The double linear interpolation uses the relative horizontal position, the relative vertical position, and the previously calculated, associated gain values.

ASTAR has the option to include a global wind updating capability in its wind forecast update. In this case, ASTAR uses time correlated ADS-B state vector and air referenced velocity reports from all surrounding aircraft to generate a local wind estimate at each aircraft's position. The estimated wind forecast is then updated in the manner previously described. This option is not used in the ATD-1 implementation due to communication constraints in the near-term national airspace system.

To exclude erroneous measured wind values which can typically occur when an aircraft is turning, a simple track-file for each aircraft is maintained for each aircraft's true ground track. If this ground track value is changing, based on the aircraft's current and previous track angle values, the aircraft's wind data are excluded from the wind forecast update. In other words, if an aircraft is turning, its wind estimation is not used in the internal forecast.

Once a new internal forecast has been generated, ASTAR selects the best altitudes for each waypoint, based on bounding the trajectory crossing altitude, to update the wind data profile in the trajectory input data. To determine if a new trajectory calculation is required due to a change in the forecast wind data, a waypoint-by-waypoint comparison between the wind data profiles of the current trajectory wind data and the internal forecast data is performed. If there is a significant difference between these data sets, then the trajectory profile wind data are updated using the internal wind forecast data and a trajectory recalculation is performed. The determination of a significant difference between these data sets is based on the following calculation for each wind-altitude point in the trajectory:

*if the difference in wind speed is greater than the variable “s”, then the difference is significant  
or*

*if the wind speed of the trajectory data is greater than s and the difference in wind angle is  
greater than the variable “a”, then the difference is significant, where*

*if the distance to go for the ownship, DTG<sub>ownship</sub>, is greater than 200 nmi, s = 5 kts and a = 10°,*

*otherwise if DTG<sub>ownship</sub> is greater than 20 nmi, s = 3 kts and a = 5°,*

*otherwise s = 1.5 kts and a = 5°.*

## Trajectory State Data Computation

The trajectory state data are the trajectory data, e.g., altitude, CAS, ground speed, and ground track, at a point on the trajectory. By design, speed and altitude changes occur linearly between TCP's as defined by the trajectory generator. Because of this, the determination of a trajectory state based on an aircraft's position is reasonably easy to calculate. First, the determination of the relative segment, i.e., between which two TCP's does the aircraft's position lie, must be calculated. For the example shown in figure 3, TCP<sub>1</sub> is the first TCP on the trajectory, which is typically a high-altitude, cruise waypoint, and TCP<sub>n</sub> is the last TCP, which is typically the runway threshold. Beginning with the first TCP segment, i.e., the segment defined by the TCP pair TCP<sub>1</sub> and TCP<sub>2</sub>, a determination is made if the aircraft's position lies angularly between the two TCP's and if so, is the orthogonal distance (fig. 4) between the aircraft's position and that segment a minimum for the trajectory? In this example, the aircraft is forward of TCP<sub>i</sub> (fig. 5), in the direction of the trajectory's ground path, and behind TCP<sub>i+1</sub> (fig. 6).

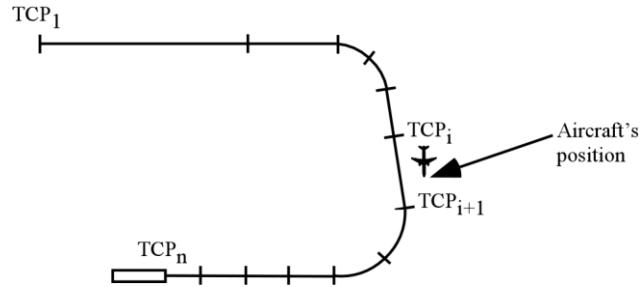


Figure 3. Trajectory state estimation.

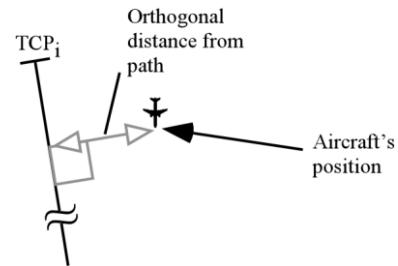


Figure 4. Orthogonal distance measurement.

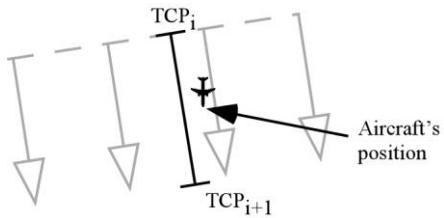


Figure 5. Aircraft's position forward of  $TCP_i$ .

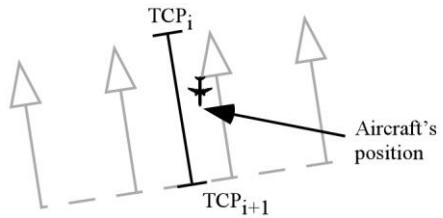


Figure 6. Aircraft's position behind  $TCP_{i+1}$

The trajectory state distance, i.e., the distance-to-go (DTG), is then simply calculated from the distance to  $TCP_{i+1}$  ( $DTG_{i+1}$ ) plus the relative distance between  $TCP_{i+1}$  and the projection of the position onto the segment (this relative distance is shown as  $d$  in figure 7). The trajectory altitude is then computed using a simple linear interpolation between the distance between the trajectory state point ( $d$  in figure 7) and  $TCP_{i+1}$  and the distance between  $TCP_i$  and  $TCP_{i+1}$ , i.e.,  $DTG_i - DTG_{i+1}$ . For example, the altitude,  $alt$ , at a position  $p_d$  on the trajectory can be calculated as:

$$x = d / (DTG_i - DTG_{i+1})$$

$$alt_d = x * alt_i + (1 - x) * alt_{i+1}$$

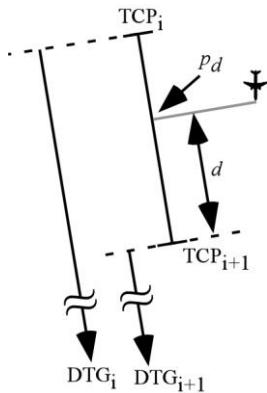


Figure 7. Distance-to-go measurement.

Since speed changes are constant between TCP's, the trajectory state speeds and time at  $p_d$  may be calculated using the linear equations of motion. For example, the CAS at  $p_d$ ,  $CAS_d$ , may be calculated as follows:

$$CAS_d = \text{square root} (x * CAS_i^2 + (1 - x) * CAS_{i+1}^2)$$

The determination of the trajectory state from the TTG can be computed using a similar technique.

## **Calculation of the Spacing Interval**

The TBO spacing interval provided by ATC may be given to ASTAR in either time or distance. An explanation of these two spacing interval types is provided in the following two sections.

### **Basic TBO Time Interval**

The basic time spacing interval is the interval that ATC would assign for the spacing aircraft to obtain at the achieve-by point against the assigned TTF. The basic spacing interval for ASTAR is a time-reference interval against a TTF that is landing on the same runway as the ownship. The operational goal in this situation is for the ownship to cross the achieve-by point at the assigned interval after the TTF crossed the achieve-by point. For this basic time interval case, there is no additional calculation required for the spacing interval; it is simply the time assigned by ATC.

### **Basic TBO Distance Interval**

In the basic distance spacing interval case, the operational goal is for the ownship to be at the ATC assigned distance behind the TTF just as the TTF crosses the achieve-by point. As in the basic time interval case, the same runway is used by both the TTF and the ownship. For this case, the distance spacing interval is converted to a time-based interval using the speeds associated with the 4D trajectory. The applicable spacing time that is used by the control law is calculated from the 4D trajectory by determining the ownship's trajectory state at the assigned spacing interval distance-to-go from the achieve-by point. The spacing time goal is then the time-to-go to the threshold at this distance. That is, the relevant spacing time is the time-to-go on the ownship's trajectory at a distance-to-go equal to the assigned spacing distance.

### **Achieve-By and Termination Point at the Final Approach Fix**

For the situation where the achieve-by point is the runway threshold, the use of an achieve-by algorithm coupled with the operational requirement to attain a stabilized approach means that the algorithm must compensate for differences in the TTF's and the ownship's actual final approach speed prior to the stabilized approach point. This capability is implicitly provided by the use of the respective aircrafts' final approach speeds in the calculation of their trajectory times to the runway threshold. For the situation where the achieve-by point at the final approach fix or beyond, the algorithm offsets the aircrafts' TTG value by the time difference between the time to the runway threshold and the time to the ABP.

If the ABP is at or farther from the runway than the FAF, as it generally is for ATD-1, once the traffic aircraft crosses the FAF, the adjusted spacing time interval for figure 8 is then calculated as:

$$\begin{aligned} \text{adjusted spacing time interval} = \\ \text{adjusted spacing time interval}_{\text{original}} - (\text{current time} - \text{ABP crossing time}_{\text{TTF}}), \end{aligned}$$

where the  $\text{adjusted spacing time interval}_{\text{original}}$  is the original ATC assigned spacing goal along with any other adjustments, e.g., runway offset and  $\text{ABP crossing time}_{\text{TTF}}$ , which is the time that the TTF crossed the ABP. Note that in the subsequent calculation of the nominal spacing time that the value for the  $\text{TTG}_{\text{TTF}}$  (fig. 8) is set to zero. A similar technique can be used in the spacing distance calculations.

## TBO Speed Control Law

The use of the trajectory calculations in the speed control law is relatively straightforward. The time error term calculation described previously,

$$t_{error} = DTG_{ownship} - t_{nominal},$$

is used in the speed control law (fig. 8). The overall design concept for this control law was to command speeds within  $\pm 15\%$  of the nominal speeds, providing some level of speed predictability to the flight crew and to ATC. This technique also eliminates the unbounded speed command problem noted in reference 14.

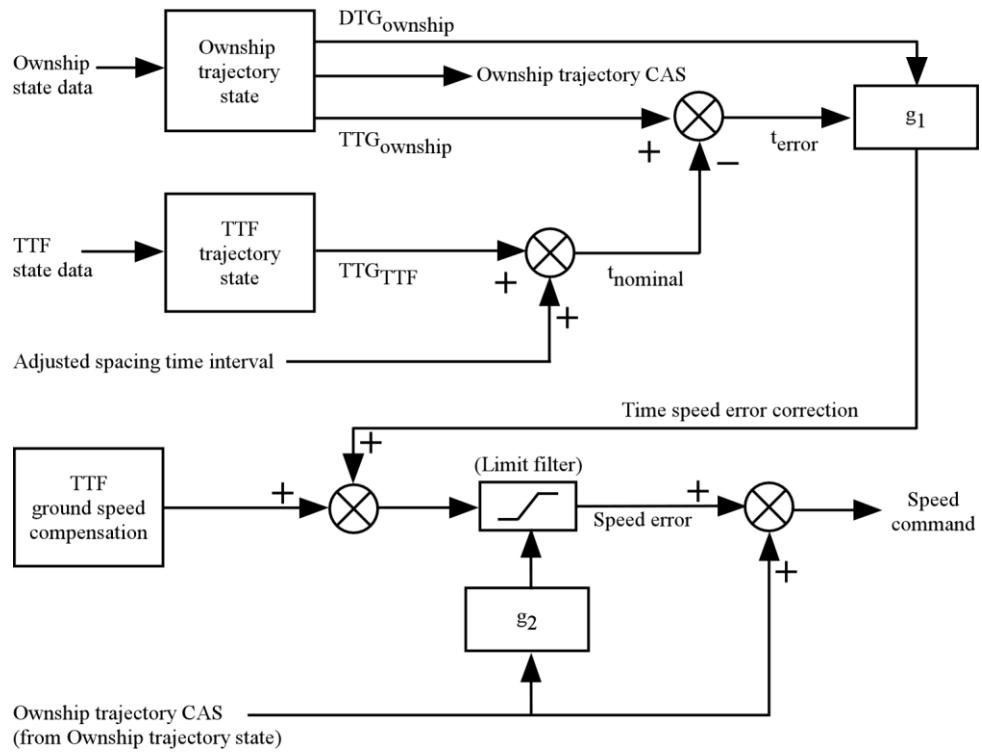


Figure 8. Speed control law.

The value of  $g_1$  (fig. 9), which is used to gain schedule the speed error term, is:

$$DTG_{ABP} = DTG_{ownship} - DTG_{ABP \text{ to end-of-path}}$$

if  $DTG_{ABP}$  is greater than 100 nmi,  $g_1 = 0.375$ ,

otherwise if  $DTG_{ABP}$  is greater than 40 nmi,  

$$g_1 = 0.375 + 0.125 * (100 - DTG_{ABP}) / 60,$$

otherwise if  $DTG_{ABP}$  is greater than 20 nmi,  

$$g_1 = 0.5 + 0.5 * (40 - DTG_{ABP}) / 20,$$

otherwise if  $DTG_{ABP}$  is greater than 5 nmi,  

$$g_1 = 1.0 + 0.5 * (20 - DTG_{ABP}) / 15,$$

otherwise  $g_1 = 1.5$ .

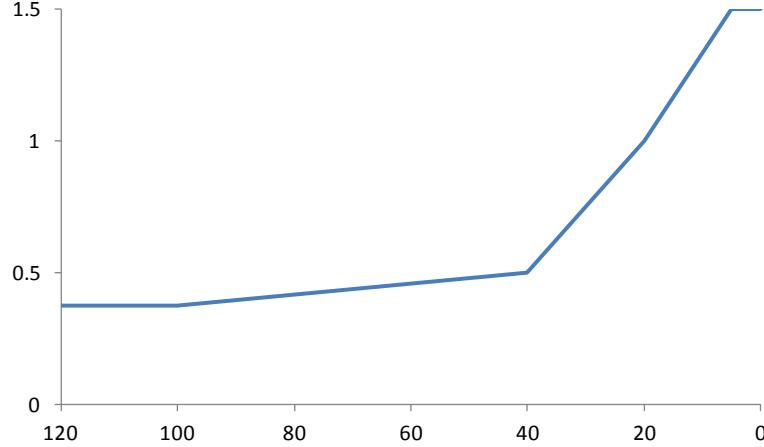


Figure 9. Gain schedule,  $g_1$ .

The value of  $g_2$  is 0.15. This limit filter limits the speed-error value to  $\pm 15\%$  of the ownship's nominal trajectory speed. At 10 nmi, these values produce 1.5 kt of speed correction (fig. 9) for one second of time error.

For the case of the RTA, the nominal spacing time is simply:

$$t_{nominal} = RTA - \text{current time}$$

where this value is substituted for the nominal spacing time from the TTF data and the TTF ground speed compensation term is set to zero in figure 8.

In ASTAR, ground speed compensation (fig. 10) is used in the TBO control law (fig. 8) as a way to limit the ownship closing too quickly on the target aircraft when both are relatively far from the airport. Ground speed compensation in this implementation only compensates for slower than nominal speeds by the TTF and was meant to compensate for the accepted ATC tendency to slow aircraft below the nominal arrival speeds when they are farther from the airport. Ground speed compensation in ASTAR does not enhance the spacing accuracy but was meant to increase controller acceptability.

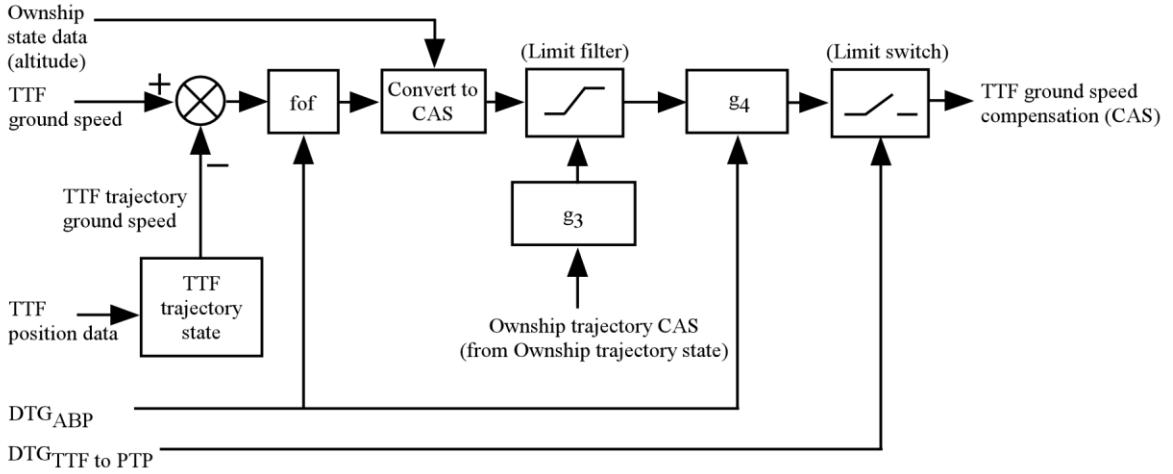


Figure 10. Ground speed compensation.

In this ground speed compensation, the difference between the TTF's actual and planned ground speeds is input into a first order filter, with this first order filter, *fof*, being used to dampen out short term variability and noise within this difference. The time constant, *t*, used in this filter is as follows:

$$DTG_{ABP} = DTG_{ownship} - DTG_{ABP \text{ to end-of-path}}$$

*if DTG<sub>ABP</sub> is greater than 30 nmi, t = 60 sec,*

*otherwise if DTG<sub>ABP</sub> is less than or equal to 0, t = 0,*

*otherwise t = 30 sec + DTG<sub>ABP</sub> (sec / nmi)*

The output of the first order filter value is then converted to a CAS term using the ownership's current altitude. This value is then gain limited, with a *g<sub>3</sub>* limit term of +0 to -0.15. This limit filter limits the compensation value to a range between +0% to -15% of the ownership's nominal trajectory speed.

Finally, a gain schedule term, *g<sub>4</sub>*, and a switch based on the TTF's distance to the planned termination point, DTG<sub>TTF to PTP</sub>, are used to eliminate the ground speed compensation input into the basic control law as the ownership approaches the runway. This is because as the aircraft approaches the runway, the spacing error becomes the critical value and the ground speed compensation becomes insignificant. The *g<sub>4</sub>* term is defined as follows:

*if DTG<sub>ABP</sub> is greater than 40 nmi, g4 = 1.0,*

*otherwise if DTG<sub>ABP</sub> is less than 20 nmi, g4 = 0,*

*otherwise g4 = (DTG<sub>ABP</sub> - 20 nmi) / (20 nmi)*

If the TTF's distance to the planned termination point, DTG<sub>TTF to PTP</sub>, is greater than 0 nmi, the *TTF ground speed compensation* value is set to the output value from the *g<sub>4</sub>* gain schedule, otherwise the value is set to 0.

Because the operational envelope for this algorithm includes high altitude Mach portions, both the trajectory calculations and the TBO control law accommodate Mach. If the aircraft is operating in a Mach

regime, then the Mach value from the trajectory data, converted to CAS, is used in the control law. The commanded CAS from the control law is then converted to a Mach command for output.

Finally, there comes a point on final approach when the ownship needs to decelerate to its final approach speed and speed changes to correct spacing errors are no longer appropriate. The earlier subsection titled ‘Final Approach Speed’ describes the two typical operational techniques for terminating active spacing and transitioning to the aircraft’s planned final approach speed. An example of how this capability is supported in ASTAR is shown in figure 11. In a purely nominal situation, i.e., where there was no spacing error, the speed command would simply follow the nominal trajectory speed profile with the deceleration to the aircraft’s final approach speed beginning at the nominal point on the trajectory. If the commanded speed were faster than the nominal speed (fig. 11), then the deceleration to the final approach speed would need to occur earlier. To accommodate this situation, ASTAR projects the final approach speed deceleration backwards from the nominal beginning of the final deceleration segment. Once the commanded speed point intercepts this deceleration line, ASTAR transitions into a final speed mode and provides a speed command that equals the appropriate speed along this deceleration line. An analogous technique is used for the situation where the commanded speed prior to the final deceleration is slower than the nominal speed (fig. 12). In this case, ASTAR would again maintain the original commanded speed until the commanded speed point intercepts this deceleration line, with the intercept point being after the nominal beginning of the deceleration segment.

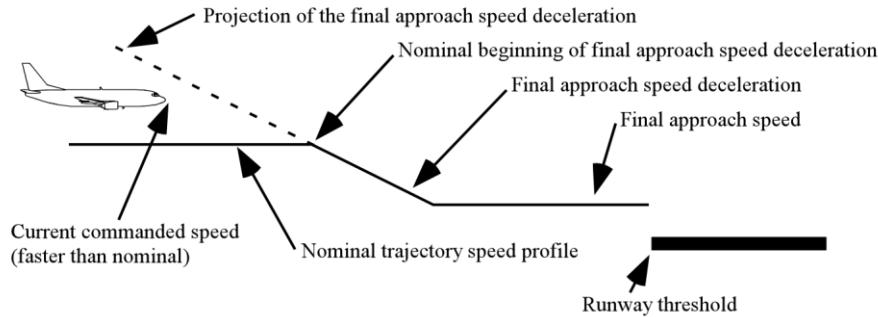


Figure 11. Final approach speed deceleration from an initially faster commanded speed.

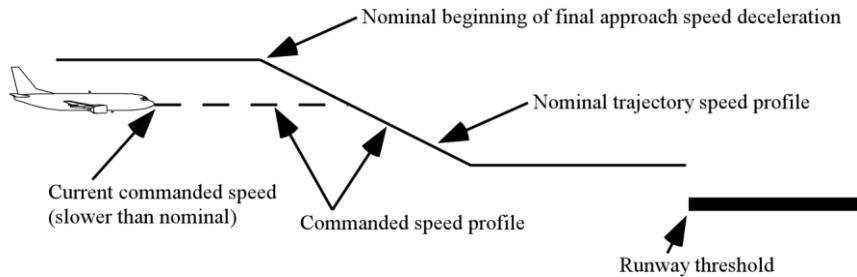


Figure 12. Commanded speed profile from an initially slower commanded speed.

## TBO Speed Change Minimization and Lag Compensation

One of the most significant design requirements for the last two versions of ASTAR is the ability to support a low cost, aircraft retrofit option with very minimal integration with other aircraft systems. In this option, it was assumed that the speed command value would be presented to the pilot and the pilot would then change the speed target of the autothrust system to match the commanded speed from ASTAR or, less likely, directly track the speed command through manipulation of the thrust levers. While this option is probably less than ideal from both a human factors and speed tracking performance perspective, there has been interest from the aviation community in providing a relatively low cost option (ref. 22) for

airborne self-spacing. To support this option, from a pilot workload standpoint it was deemed beneficial to minimize the number of speed command changes presented to the pilot. Several capabilities are provided within the algorithm that attempt to balance the number of speed changes against the spacing performance. The implementation of these capabilities has led to a considerable increase in complexity of the ASTAR algorithm.

### **TBO End Speed Estimation**

In this implementation of ASTAR, the pilot is expected to implement the algorithm's speed command by matching the aircraft's autothrust command to the ASTAR speed command. During a programmed deceleration segment without any spacing error, e.g., a change in the nominal speed profile from 210 kt to 170 kt (fig. 13), the ASTAR speed command would change continuously during the deceleration segment, with the command speed following the nominal speed profile. To reduce pilot workload so that the pilot did not need to continuously monitor the speed command and continuously change the input to the autothrust system, a secondary speed command is output by ASTAR for display to the pilot. This secondary speed command, termed the end speed command, is an estimate of the speed command at the end of the speed change. In the example of figure 13, the end speed command would change from 210 kt to 170 kt as the aircraft reaches the start of the 210 to 170 kt deceleration segment. For long deceleration segments, the end speed command could be used first by the pilot to set the autothrust speed target and then the basic, instantaneous speed command could be used to modulate the thrust or aircraft's drag devices to better follow the decelerating speed command profile.

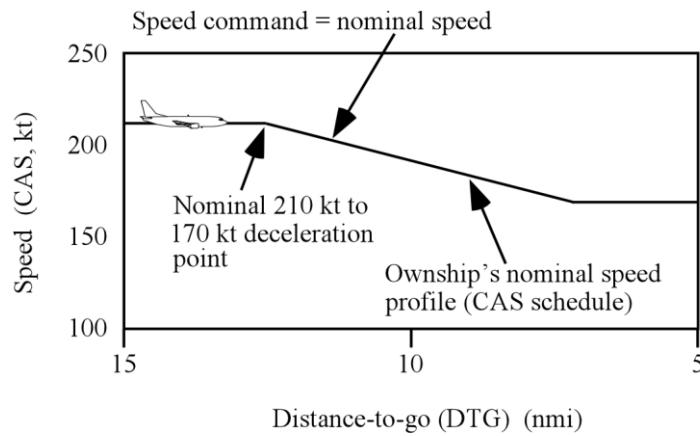


Figure 13. Example speed change with no spacing correction.

A similar situation would occur in the presence of a required speed correction due to a spacing error or RTA adjustment. In figure 14, the nominal speed profile is the same as in figure 13, but there is now a positive 10 kt spacing correction. Prior to the start of the nominal 210 to 170 kt deceleration segment, both the speed command and the end speed command would be 220 kt. At the start of the deceleration segment, the speed command would be 220 kt while the end speed command would change to 180 kt.

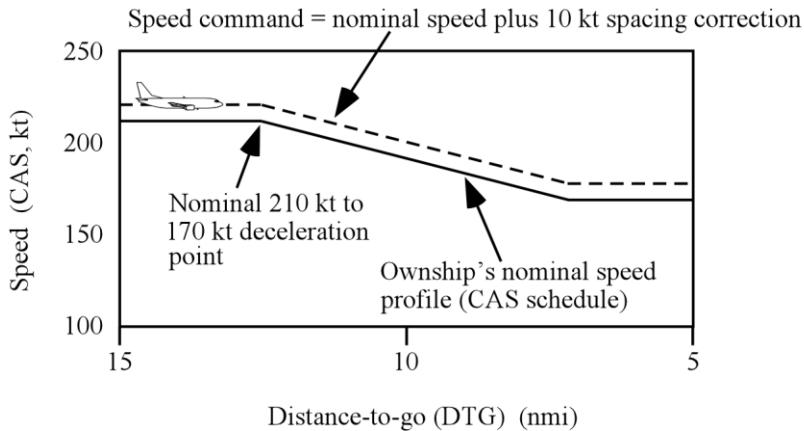


Figure 14. Example speed change with a positive 10kt spacing correction.

### **TBO Speed Command Quantization**

Another means for reducing the number of speed changes in ASTAR was to use a quantization technique on the end speed command and, except during speed changes, on the instantaneous speed command. By applying a quantization to the speed command prior to its output, the end speed command changes only occur in discrete intervals, thus reducing the number of commanded speed changes. For example, if the speed command (fig. 8) was to change from 210 kt to 172 kt and a 5 kt quantization value was used, then the following would occur:

- Immediately prior to the speed change, the output values for both the speed command and the end speed command would be 210 kt.
- At the start of the speed change, the output value for the speed command would slowly begin to decrease, e.g., 209, 208, 207 kt. The output value for the end speed command, because it is being "chunked" in 5 kt increments by the quantization process, would change to 170 kt.
- At the end of the speed change, the output values for both the speed command and the end speed command would be 170 kt.

Hysteresis was included in the quantization logic to reduce dithering of the end speed command when the command speed is near the breakpoint for the quantization value.

### **TBO Nominal Deceleration Roll-In Logic**

During the initial evaluation of ASTAR10 (ref. 23), it was determined that the lag in response to a speed command change by the simulated aircraft was problematic and contributed to undesirable spacing performance, especially under situations where several aircraft were spacing one after another, i.e., a spacing string. To reduce this problem at the start of a planned deceleration segment in the nominal profile, where this response lag was most apparent, predictive, nominal speed roll-in logic was added to the speed command. An example of a deceleration in the nominal profile without this roll-in logic is shown in figure 15. In this example, there is no speed error, so the instantaneous speed command would match the nominal speed profile. Additionally, the change in the end speed command would occur at the deceleration point on the nominal speed profile. Therefore, at 300 seconds TTG in the example of figure 15, the end speed command would change from 210 kt to 170 kt and the instantaneous speed command would begin to decrease at a rate equal to the change in the nominal speed profile. Using a 12 second look ahead, the equivalent situation is shown in figure 16 with the roll-in logic. In figure 16, the end speed command would change from 210 kt to 170 kt 12 seconds earlier relative to the basic profile. In this

situation (fig. 16), the instantaneous speed command would change in a manner such that the nominal deceleration rate and speed would just match the nominal command speed and deceleration rate 24 seconds after the start of the roll-in period.

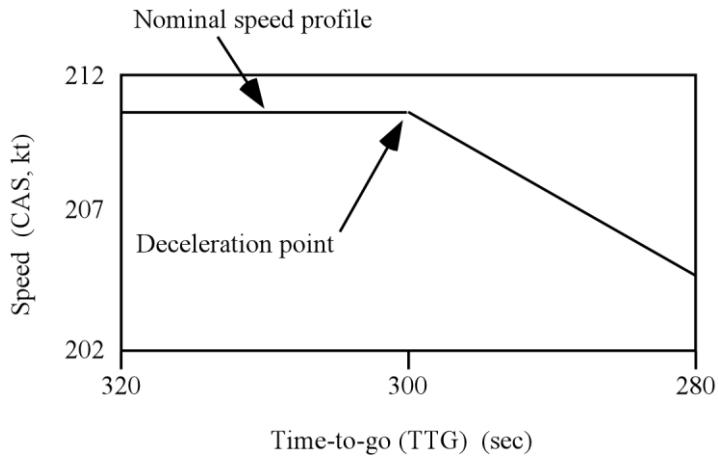


Figure 15. Nominal speed change without roll-in logic.

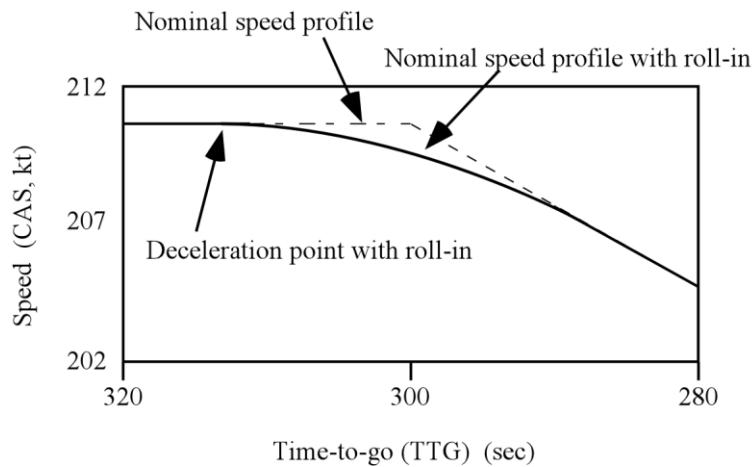


Figure 16. Nominal speed change with roll-in logic.

#### ***Look Ahead Speed Change Inhibit***

To minimize the number of speed changes prior to a programmed deceleration segment, i.e., where the planned trajectory specifies a deceleration, a look-ahead speed change inhibit option was used. In this regard, the algorithm would look ahead by 10 seconds in the nominal speed profile (fig. 17) to determine if a change onto a deceleration segment would occur. Within this 10 second interval, any speed command increase would be inhibited. If the nominal deceleration roll-in logic, described in the previous section, is used, its 12 second roll-in interval would be added to the 10 second look-ahead interval. Thus, the speed change inhibit logic would be applied 22 seconds prior to the deceleration point on the basic, nominal speed profile.

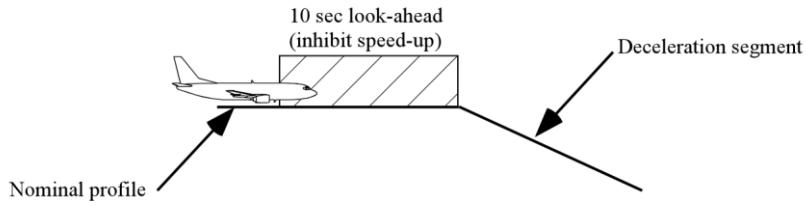


Figure 17. Look-ahead speed-up inhibit.

## Stated-Based, Constant-Time Delay Algorithm

The fundamental concept used for the state-based algorithm in ASTAR13 was derived from previous work at NASA Langley (refs. 13, 15, and 16). The state-based algorithm in ASTAR13 is a type of time-history algorithm referred to as a constant-time delay (CTD) algorithm. The CTD algorithm was designed to minimize the measured spacing interval, which is defined as the difference in time between when the traffic aircraft crossed the ownship's current along-path position and the current time. This fundamental concept is shown in figures 18 and 19. In these figures, the lead or traffic aircraft is ahead of the FIM or ownship aircraft. The traffic aircraft is continuing to reduce its speed and the assigned spacing goal is such that the FIM aircraft is behind its nominal position. In figure 19, the state-based, CTD algorithm would, for this situation, calculate the spacing error, calculate an appropriate speed correction value (i.e., the speed correction due to spacing error), and add this correction value to the traffic's speed history value that occurred at the ownship's current position. This new value would be the resulting intermediate speed command.

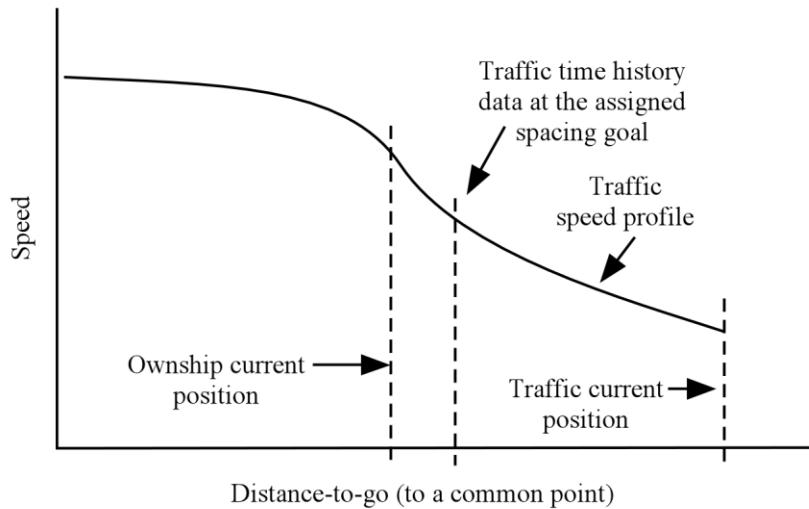


Figure 18. CTD nominal profile data.

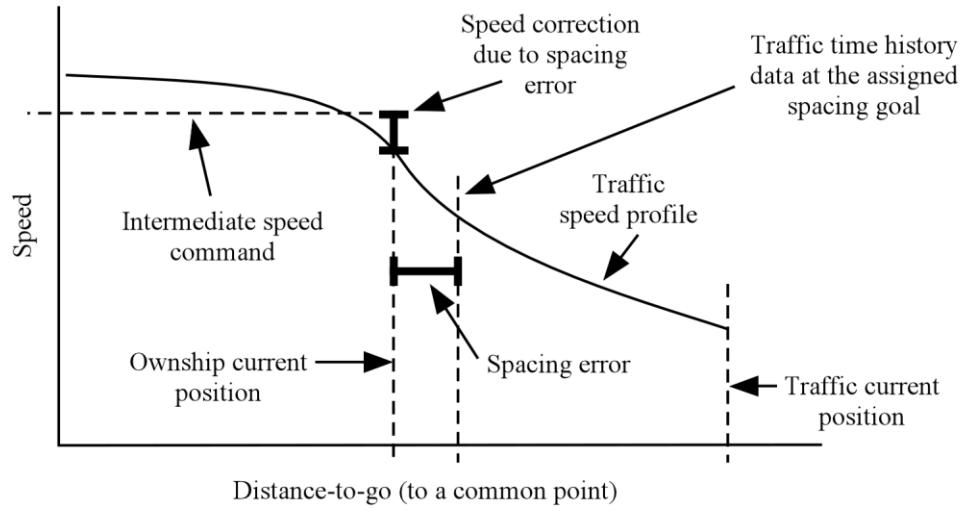


Figure 19. CTD spacing error and speed correction.

This latest adaptation of the CTD algorithm includes several new innovations that better support the near-term operational environment and the RTCA FIM MOPS. A flow diagram of this CTD algorithm is shown in figure 20. The subsequent text will briefly describe several general design considerations for each block in this diagram.

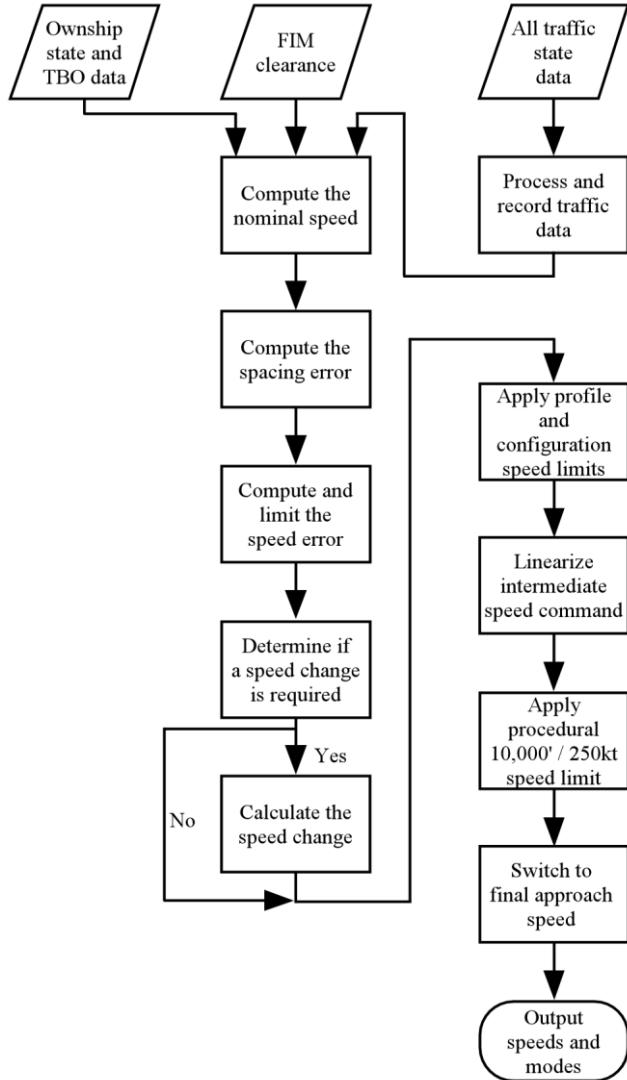


Figure 20. CTD block diagram.

## CTD Design Considerations and General Implementation Notes

One of the more significant differences between the previous versions of ASTAR and the CTD portion of ASTAR13 is the derivation of when speed changes occur. All of the previous versions of ASTAR and the TBO portion of this latest implementation assumed in the basic design that the actual speed control would be conducted in a manner such that ASTAR would be directly driving the aircraft's autothrust system, i.e., that the ASTAR speed command would be directly integrated in some manner with the aircraft's speed control system. Because of this, what is considered the intermediate or instantaneous speed command is continually calculated, with this speed command continuously adjusting the autothrust command. Since the initial development of ASTAR, however, the industry focus and operational testing has moved toward a low cost solution with the FIM equipment no longer expected to be integrated with the aircraft systems. Because of this move from an integrated solution toward a federated, stand-alone implementation, an additional speed command was added to the last two versions of ASTAR (see the *End Speed Estimation* section in this document). The additional speed command was the end commanded speed, which is the projected or expected speed command at the end of a change in the commanded speed. The end commanded speed is expected to reduce pilot workload in the stand-alone implementation if it is changed in discrete increments (see the *Speed Command Quantization* section in this document). It

is the determination of the point at which the end command speed quantization (end command speeds in discrete increments) occurs and the calculation of the CTD algorithm's end commanded speed that is innovative in this solution. These capabilities will be further described in the speed change description sections. A representative plot of the end commanded speed is given in figure 21. This may be compared with the plot of intermediate speed command given in figure 19.

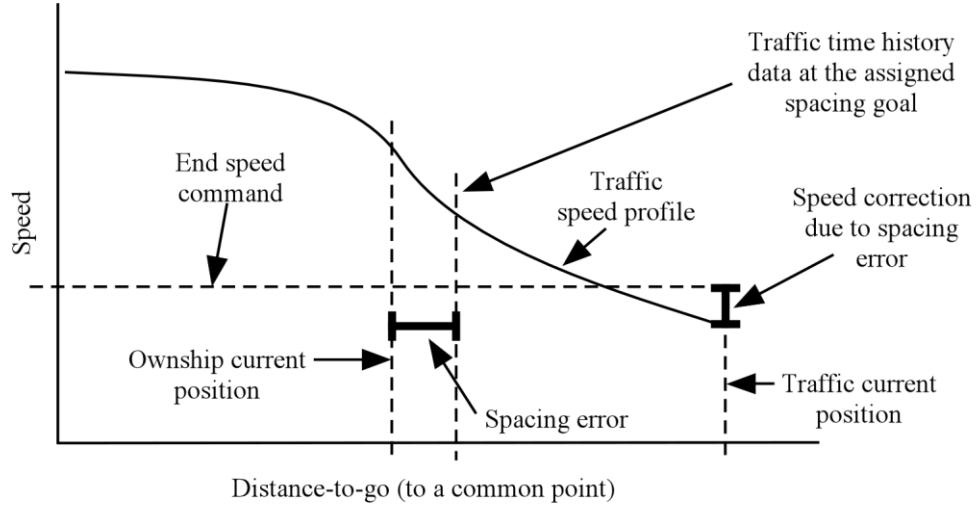


Figure 21. CTD end speed command.

A conversion process that was found to be beneficial during early testing of the CTD algorithm is in regard to how the traffic's ground speed is both averaged and converted to airspeed (CAS). Because the resulting speed commands are required to be CAS values, attention must be taken in the conversion of the ground speeds derived from the traffic's ADS-B data into a CAS values for use by the control law. All ground speed to CAS conversions used by the CTD algorithm use the following method, where the ground speed is from a position  $p$  in the traffic's time history data:

$$CAS = CAS \text{ conversion}(\text{ averaged ground speed}_p + headwind_p, altitude, temperature),$$

where CAS conversion is a standard true airspeed to CAS conversion; *averaged ground speed* is a derived and averaged ground speed value from the traffic's time history data at position  $p$  (see section *Process and Record Traffic Data*); the  $headwind_p$  is the current wind speed for the ownship, adjusted for the difference between the current wind angle and the traffic's ground track angle at position  $p$ , such that:

$$\begin{aligned} headwind_p &= \text{ownship's current wind speed} * \\ &\quad \cosine(\text{ownship's current wind angle} - \text{traffic's ground track at } p); \end{aligned}$$

$$\text{altitude} = \text{altitude}_p + \text{altitude bias},$$

where  $\text{altitude}_p$  is the altitude value from the traffic's time history data at position  $p$  and the *altitude bias* is the ownship's altitude minus the traffic's altitude (ownship's altitude - traffic's altitude) at the ownship's position; and *temperature* is the air temperature for the ownship.

Data values for variables are retained between iterations of the CTD algorithm. If the initial values for these data values are not explicitly described, then it should be assumed that they are initialized to some

reasonable value. Because of the complexity in describing several sections of this CTD algorithm, pseudo-code is included to enhance understandability. Nesting levels in the pseudo-code description are denoted by the level of indentation of the document formatting. Additionally, long sections of logic may end with *end of statements* to enhance the legibility of the text. Key data variables for the CTD in the pseudo-code descriptions are:

*CAS<sub>1</sub>*: the intermediate speed command prior to any limiting or quantization.

*CAS<sub>2</sub>*: a quantized value of *CAS<sub>1</sub>*.

*CAS Step Size*: the quantization size or grain size in the quantization of the command speed.

*Deceleration Start Speed*: the speed at the start of a deceleration segment.

*End Speed Command*: the output end speed command that includes speed limiting.

*Interim Speed Command*: the interim, intermediate speed command prior to speed limiting.

*Interim End Speed Command*: the interim, end speed command prior to speed limiting.

*In Deceleration*: a status variable indicating that the changes to the intermediate speed command are occurring along a decelerating portion of the traffic's speed history profile.

*Nominal History Speed*: the CAS value used as the current nominal speed in the calculation of the *Interim Speed Command*.

*Saved CAS<sub>1</sub>*: the *CAS<sub>1</sub>* value from the previous speed change.

*Speed Command*: the output intermediate speed command that includes speed limiting.

*Speed Error*: the CAS correction that is applied to the nominal or profile CAS to calculate the intermediate speed command.

A discussion for using a distance-based clearance is presented at the end of this section. Unless otherwise stated, the subsequent text addresses time-based clearance.

## **Ownership state and TBO data**

The ownership state data and trajectory data are inputs from the imbedded TBO algorithm. Valid TBO data are required for variables such as the DTG to the planned termination point and the nominal TBO profile CAS to the ownership's current position.

## **FIM Clearance**

The FIM clearance information, specifically the assigned spacing goal (ASG) which includes the spacing dimension, i.e., time or distance.

## **ALL Traffic State Data**

The RTCA FIM MOPS allows at the reception of the FIM clearance that any traffic time history data older than the time at FIM clearance initiation may be extrapolated from the traffic's current ADS-B record. This technique, however, will initially introduce large speed command errors if the traffic has not been in constant altitude, constant steady-state flight. One of the innovations in ASTAR13 is that it

continuously monitors and records the traffic records for all of the surrounding aircraft. At the initiation of the FIM clearance, the algorithm locates the basic ADS-B data for the aircraft that is now specified as the traffic aircraft in the FIM clearance and populates the CTD traffic history data records with these time-based data sets.

## **Process and Record Traffic Data**

The CTD algorithm requires time history data from the traffic aircraft in order to calculate the measured spacing interval and calculate the speed that the traffic aircraft had when it was at the ownship's current position. To ensure that the traffic aircraft's time-history information is available when a new IM clearance initiates, ASTAR13 stores time history information for all aircraft that are broadcasting ADS-B data. Traffic data are normally stored at the reception of new ADS-B data, which occurs at approximately 1 Hz. In addition to the traffic's basic state data (data time, latitude, longitude, altitude, ground speed, and ground track), a pseudo distance-flown and an averaged ground speed variable are added to each record. The pseudo distance-flown variable is initialized to 0, with the distance correlated to the first, oldest data point for the traffic. The averaged ground speed variable is an averaging of the past 4 seconds of ground speed data. In lieu of the basic ground speed value, this value is used for all of the ground speed computations in ASTAR13 requiring the traffic's ground speed.

## **Compute the Nominal Speed**

Previous CTD concepts computed the speed command as:

$$\text{Intermediate Speed Command} = \text{Nominal History Speed} + \\ \text{speed correction due to the spacing error},$$

where the *Nominal History Speed* is the traffic's speed history at the ownship's current position. What was found in previous ASTAR experiments, however, was that some speed command anticipation was required to overcome the time lag in the ownship's response to a change in the speed command. Based on previous experimental data, a 15 second lead compensation was found to be adequate. The computation of the *Nominal CAS History Speed* is then:

$$\text{Nominal CAS History Speed} = \text{CAS conversion}(p),$$

where  $p$  is traffic's time history data from 15 seconds in front of the ownship's current position.

## **Compute the Spacing Error**

The time spacing error is simply the difference between the MSI and the spacing goal:

$$SE_{time} = MSI_{time} - ASG_{time},$$

where the MSI is computed as

$$MSI_{time} = \text{current time} - \text{time when the lead aircraft was at the FIM aircraft's proximate position}.$$

## **Compute and Limit the Speed Error**

The calculation of the speed error value is based on the spacing error, SE, and the gain schedule is based on the FIM aircraft's distance to the planned termination point (PTP) and the magnitude of the spacing error. This gain scheduling, as in the TBO portion of ASTAR13, is designed to reduce the sensitivity of the control law when the FIM aircraft was far from the PTP but provide a highly increased sensitivity when close to the PTP. This sensitivity adjustment is designed to reduce the number of speed command changes while providing high spacing accuracy at the PTP and ensuring that the spacing error

is maintained within the required tolerance once captured. The actual gain scheduling logic contains heuristics to eliminate gain value flip-flopping at the schedule break points. The basic gain values are:

```

if ( $|SE| > 30 \text{ sec}$ ) then gain = 0.5
otherwise if ( $|SE| > 10 \text{ sec}$ ) then
    if (the FIM aircraft's distance to the PTP > 5 nm) gain = 0.5
    otherwise gain = 1.0
otherwise
    if (the FIM aircraft's distance to the PTP > 5 nm) gain = 1.0,
    otherwise gain = 1.5

```

The speed error value is then:

$$\text{Speed Error} = \text{gain(kt/sec)} * SE.$$

To prevent saturating the calculated speed command, this value for *Speed Error* is then limited to  $\pm 33\%$  of the *Nominal CAS History Speed*.

### Determine if a Speed Change is Required

In the ASTAR13 CTD algorithm, as in the TBO portion, a significant part of the design was focused on minimizing the number of speed command changes while maintaining an accurate spacing interval. A large part of this design feature occurs in the determination of the need for a speed change, and if a speed change is not required, either provides a smooth transition to or maintains the previously calculated command speed. This process of only changing the speed command if that change is larger than some threshold value, relative to the previous command, is an innovative feature of ASTAR13. The following process is used to determine the speed change requirement:

The quantization value, used in the majority of the subsequent CAS calculations, is:

```

if (FIM aircraft's time-to-go to the PTP > 60 sec) then CAS Step Size = 10
otherwise CAS Step Size = 5
if (In Deceleration ≠ true) then
    CAS1 = Nominal CAS History Speed + Speed Error

```

Determine the direction of the difference.

```

if (CAS1 > Saved CAS1) then Up = true
otherwise Up = false,

```

Quantization test to determine if there is a change using a 5 kt directional bias if the direction of change and the error are in opposite directions.

*if ((Up = true) and (Speed Error < -5 kt)) then*

$CAS_2 = round((CAS_1 - 5 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$

*otherwise if ((Up = false) and (Speed Error > 5 kt))*

$CAS_2 = round((CAS_1 + 5 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$

*otherwise*

Use a 2 kt offset to prevent flip-flopping.

*if (Up = true) then  $CAS_2 = round((CAS_1 - 2 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$*

*otherwise  $CAS_2 = round((CAS_1 + 2 \text{ kt}) / \text{CAS Step Size}) * \text{CAS Step Size}$*

Now compare the  $CAS_2$  value with the previously calculated interim value for the intermediate speed command, *Saved CAS<sub>2</sub>*, to determine if a speed change is needed.

*if (((Up = true) and ( $CAS_2 > \text{Saved CAS}_2$ )) or*

*((Up = false) and ( $CAS_2 < \text{Saved CAS}_2$ ))) then Speed Change = true*

*otherwise Speed Change = false*

*end of statement if (In Deceleration ≠ true)*

## Calculate the Speed Change

If a speed change is required, based on the state of the *Speed Change* variable determined in the previous subsection, then values for both the interim intermediate speed command, *Interim Speed Command*, and the interim end speed command (see the section *End Speed Estimation*), *Interim End Speed Command*, are calculated. If a speed change is not required, then the previously computed values are retained. The speed command values are computed as follows:

*if ((In Deceleration ≠ true) and (Speed Change)) then*

*Saved CAS<sub>1</sub> = CAS<sub>1</sub>*

Save the speed value before it is changed.

*Deceleration Start Speed = Interim Speed Command*

*Interim Speed Command = CAS<sub>2</sub>*

*Interim End Speed Command = CAS<sub>2</sub>*

Determine if a large deceleration is occurring and if so, estimate the end speed command at the end of the deceleration. This deceleration evaluation process is an innovation designed to minimize the number of speed commands presented to the flight crew.

*if (Up = false) then*

Beginning at the same point used in *Compute the Nominal Speed* for determining the *Nominal History Speed* value, i.e., 15 seconds in front of the ownship's position, determine if the changes in the traffic's history average ground speed values indicate that a deceleration by the traffic aircraft is occurring. If so, determine the point in the traffic's history data when the deceleration ends (fig. 22).

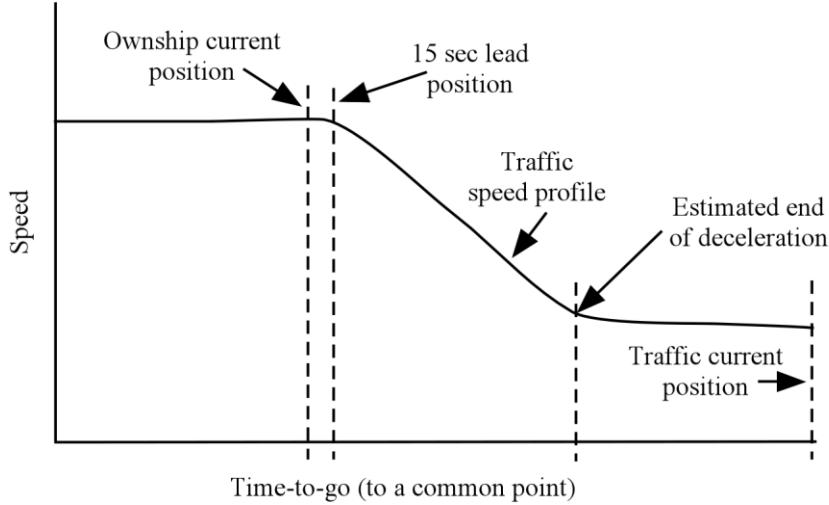


Figure 22. CTD end of deceleration estimation.

This deceleration estimation uses the following process:

The test conditions are initialized such that the data record position in the traffic's history record,  $p$ , is set to the 15 second lead position and the end of the deceleration CAS value, *Deceleration End CAS*, is calculated using the previously described CAS conversion at position *Deceleration Position*.

*Valid Deceleration = false*

*i = Deceleration Position*

*last = Deceleration Position*

*Finished = false*

*iterate until (Finished = true)*

*i = i + the position 4 history records closer to the traffic's current position*

*if(i > traffic's current position) then Finished = true*

*otherwise*

*Test CAS = CAS conversion(i)*

If the difference between the current test CAS value and the previous minimum CAS value is greater than an average of 0.8 kt / sec, i.e., the traffic's speed has increased, then the deceleration has ended.

*if (Test CAS > (Deceleration End CAS + 0.8 \* (traffic history record time at position i - traffic history record time at position last))) then Finished = True*

If the current test CAS value is 0.333 kt / sec slower than the previous minimum CAS value, then the deceleration has started or is continuing.

*if (Test CAS < (Deceleration End CAS - 0.333 \* (traffic history record time at position i - traffic history record time at position last))) then*

*Valid Deceleration = true*

*Deceleration Position = i*

*Deceleration End CAS = Test CAS*

*last = i; this is the end of the deceleration estimation process.*

If a profile deceleration exists, then initialize the variables used to calculate the interim speed command during the deceleration.

*if (Valid Deceleration = true) then*

*In Deceleration = true*

*Deceleration Start Time<sub>1</sub> = current clock time*

*Deceleration Start Time<sub>2</sub> = traffic's history data time at ownship's proximate position*

*Saved CAS<sub>1</sub> = Deceleration End CAS + Speed Error*

*end of statement if (Up = false) then*

If *In Deceleration*, calculate the interim speed values.

*if (In Deceleration = true) then*

Calculate the deceleration initial values at the start of the deceleration.

*if (Deceleration Start Time<sub>1</sub> = current clock time) then*

*s<sub>1</sub> = Deceleration End CAS + Speed Error*

*s<sub>2</sub> = round((s<sub>1</sub> + 2 kt) / CAS Step Size) \* CAS Step Size*

*Interim End Speed Command = s<sub>2</sub>*

Calculate the deceleration time interval.

$t = \text{traffic history time at Deceleration Position} - \text{Deceleration Start Time}_2$

Calculate the required deceleration rate,  $r$ .

*if* ( $t > 0$ ) *then*

$r = (\text{Deceleration Start Speed} - \text{Interim End Speed Command}) / t$

Limit to at least 0.5 kts / sec.

*if* ( $r < 0.5$ ) *then*  $r = 0.5$

$d = (\text{current clock time} - \text{Deceleration Start Time}_1) * r$

*Interim Speed Command* = *Deceleration Start Speed* -  $d$

Limit the calculated speed so that it is no slower than the end speed command.

*if* (*Interim Speed Command* < *Interim End Speed Command*) *then*  
*Interim Speed Command* = *Interim End Speed Command*

Has the end of the deceleration time been reached?

*if* (*current clock time*  $\geq$  (*Deceleration Start Time*<sub>1</sub> +  $t$ )) *then*

*In Deceleration* = *false*

*Saved CAS*<sub>1</sub> = *Interim End Speed Command*

*otherwise*, the case where  $t \leq 0$

*Interim Speed Command* = *Interim End Speed Command*

*In Deceleration* = *false*

*end of statement if* (*In Deceleration* = *true*)

*end of statement if* ((*In Deceleration*  $\neq$  *true*) *and* (*Speed Change*))

*Speed Command* = *Interim Speed Command*

*End Speed Command* = *Interim End Speed Command*

## Apply Profile and Configuration Speed Limits

The CTD speed commands, *Speed Command* and *End Speed Command*, may be limited by various speed restrictions, including procedural and airframe limitations. The limiting factors in this algorithm include:

### Limits based on the nominal profile speeds

The CTD speed commands are limited by the nominal profile speeds. If the FIM aircraft has not passed the first CAS constrained waypoint on its route, then the TBO speed information is not considered to be

valid. In this case, both the *Speed Command* and the *End Speed Command* values may be limited such that they remain within  $\pm 10\%$  of the *Nominal CAS History Speed* of the traffic aircraft. If this limitation is applied, then the resulting *End Speed Command* value will be quantized using *CAS Step Size*.

For the case where the TBO speed data are valid, both the *Speed Command* and the *End Speed Command* values may be limited such that they remain within  $\pm 15\%$  of the related TBO nominal profile speeds. For the *Speed Command*, its value is limited to  $\pm 15\%$  of the ownship's current trajectory CAS. For the *End Speed Command*, its value is limited to  $\pm 15\%$  of the ownship's projected trajectory CAS at the end of a planned deceleration, if one exists, or the ownship's current trajectory CAS otherwise. If this latter limitation is applied, then the resulting *End Speed Command* value will be quantized using *CAS Step Size*.

Last, to support the RTCA FIM MOPS requirement for closely matching the traffic's speed near the planned termination point, if the FIM aircraft is within 30 seconds of the planned termination point, the  $\pm 10\%$  of the *Nominal CAS History Speed* limitation is applied in the previously described manner.

### **Airframe limits**

Airframe limits such as the maximum operating limit,  $V_{mo}$ , or maximum flaps limits may be input in ASTAR13. If airframe limits are used, then both the *Speed Command* and the *End Speed Command* values may be limited by these limit values. If these limitations are applied, then the resulting *End Speed Command* value will be quantized using *CAS Step Size*.

### **Procedural limits**

Procedural limits are imposed for RNAV turns that include maximum speeds. If either the *Speed Command* or the *End Speed Command* is projected to be above the RNAV speed limit for a waypoint, based on the maximum of the current deceleration rate or a 0.75 kt / sec deceleration, then a speed reduction is implemented so that the speed at the waypoint does not exceed the speed limit. In this situation, the *End Speed Command* is set to the RNAV speed limit and the *Speed Command* is linearly reduced from the speed at the initiation of this deceleration to the limit speed at the waypoint.

## **Linearize Intermediate Speed Command**

To aid the pilot in transitioning to a new commanded speed, the intermediate speed command, *Speed Command*, is designed to provide a smooth and continuous series of speed commands as the algorithm transitions between different values of *End Speed Command*. As such, the *Speed Command* may be modified under the following two conditions:

### **Condition 1**

If the *Speed Command* is not equal to the *End Speed Command* and the value of the *Speed Command* is not changing, then apply a change at a constant rate. This process would be performed as follows:

Assume that  $time_{previous}$  is the time at the previous iteration of the algorithm,  $time$  is the time at the current iteration of the algorithm,  $Speed\ Command_{previous}$  is the value of the *Speed Command* at the previous iteration of the algorithm, and  $Speed\ Command$  is the current value of the intermediate speed command. Then,

*if ((Speed Command  $\neq$  End Speed Command) and  
(Speed Command<sub>previous</sub> = Speed Command)) then*

change the Speed Command by 1 kt / sec starting with the slowing condition.

$\text{if } (\text{Speed Command} > \text{End Speed Command}) \text{ then}$   
 $\quad \text{Speed Command} = \text{Speed Command} - 1.0 * (\text{time} - \text{time}_{\text{previous}})$   
 Limit any undershoot.  
 $\text{if } (\text{Speed Command} < \text{End Speed Command}) \text{ then}$   
 $\quad \text{Speed Command} = \text{End Speed Command}$   
*otherwise*  
 $\quad \text{Speed Command} = \text{Speed Command} + 1.0 * (\text{time} - \text{time}_{\text{previous}})$   
 Limit any overshoot.  
 $\text{if } (\text{Speed Command} > \text{End Speed Command}) \text{ then}$   
 $\quad \text{Speed Command} = \text{End Speed Command}$

## Condition 2

If the *Speed Command* is not equal to the *End Speed Command* and the value of the *Speed Command* has changed by more than 2 kt / sec, then apply a change at a smaller rate. This process is to eliminate impulse changes in the speed and would be performed as follows:

As previously, assume that  $\text{time}_{\text{previous}}$  is the time at the previous iteration of the algorithm,  $\text{time}$  is the time at the current iteration of the algorithm,  $\text{Speed Command}_{\text{previous}}$  is the value of the *Speed Command* at the previous iteration of the algorithm, and *Speed Command* is the current value of the intermediate speed command. Then,

$$\text{rate} = (\text{Speed Command} - \text{Speed Command}_{\text{previous}}) / (\text{time} - \text{time}_{\text{previous}})$$

*if* ( $\text{rate} < 2$ ) *then*

Change the Speed Command by 1 kt / sec starting with the slowing condition.

$$\text{Speed Command} = \text{Speed Command} - 1.0 * (\text{time} - \text{time}_{\text{previous}})$$

Limit any undershoot.

*if* ( $\text{Speed Command} < \text{End Speed Command}$ ) *then*

$$\text{Speed Command} = \text{End Speed Command}$$

*otherwise*

$$\text{Speed Command} = \text{Speed Command} + 1.0 * (\text{time} - \text{time}_{\text{previous}})$$

Limit any overshoot.

*if* ( $\text{Speed Command} > \text{End Speed Command}$ ) *then*

*Speed Command = End Speed Command*

#### **Apply Procedural 10,000 ft / 250 kt Speed Limit**

The 250 kt speed limit below 10,000 ft is implemented such that once the ownship is projected to descent below 10,000 ft within the next 60 seconds, a limiting process is applied if the command speeds crossing 10,000 ft are projected to be above 250 kt. If this condition does exist, the *Speed Command* is linearly ramped such that it reaches 250 kt at the 10,000 ft projection point. The *End Speed Command* value is immediately set to 250 kt.

#### **Switch to Final Approach Speed**

This uses the same logic that is in the previously described TBO algorithm to determine when to terminate the spacing control law and switch into the final deceleration speed logic. If the switch is made to the final approach speed logic, then the *Speed Command* and *End Speed Command* values will be modified to satisfy the final approach speed requirements.

#### **Distance-Based Clearance**

For a distance-based clearance, the state-based algorithm becomes simpler. A fixed distance-based spacing interval is effectively a station-keeping application. It can be thought of as two aircraft connected by a string, with the string length being equal to the desired spacing distance. In this situation, when the lead aircraft changes speed, the FIM aircraft would make the exact corresponding speed change. For the ASTAR13 implementation, the difference between the time-based spacing and the distance-based spacing are keyed around two critical parameters: the measured spacing interval and the *Nominal History Speed*. For the time-based clearance, the MSI is the difference between the current time and the time when the lead aircraft was at the FIM aircraft's proximate position. The spacing error, *SE*, is then the difference between the MSI and the assigned spacing goal. The *Nominal History Speed* is based on the traffic's speed at a position 15 seconds in front of the ownship's position (fig. 22). For distance-based spacing, the MSI is simply the along-path distance between FIM aircraft and the lead aircraft, i.e., the along-path difference between the ownship's current position and the traffic's current position. The spacing error, *SE*, is the difference between the MSI and the assigned spacing distance goal. The *Nominal History Speed* is simply the CAS conversion of the traffic's current ground speed (fig. 23).

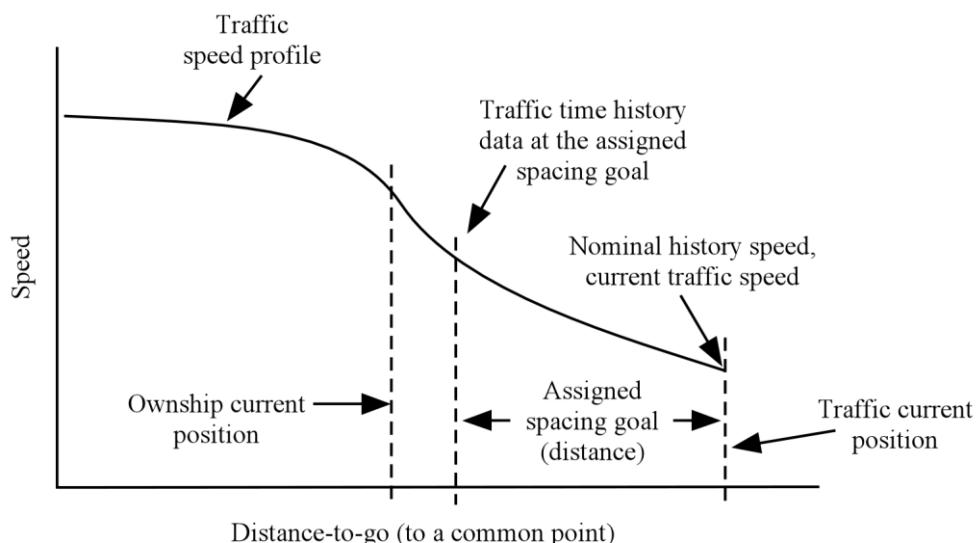


Figure 23. CTD nominal speed.

To calculate the speed commands, the distance-based variables are used in the previously described state-based algorithm. Since the *Nominal History Speed* is at the traffic's current position, there are no data to support deceleration estimation, so this part of the algorithm is not invoked.

## Control Law Selection

Figure 24 describes the control law selection logic that is used to determine whether the TBO or the CTD control algorithm is used to calculate the speed guidance. It is assumed that data values for variables are retained between iterations and that the variable *Using CTD* is initialized to false.

A description of the input variables and logic for the control law selection process are as follows:

*FIM mode*: If the overall control is one of the various FIM clearance options or an RTA.

*Achieve-by and Maintain clearance*: If the FIM clearance is the Achieve-by and Maintain option.

*Ownship valid*: The ownship's state and trajectory data are valid and the aircraft is on its planned path.

*TTF valid*: The TTF's state and trajectory data are valid and the aircraft is on its planned path.

*Ownship past first CAS*: The ownship is past the first procedurally CAS constrained waypoint on its route. This option is required to support the ATD-1 operational environment since it is assumed that the algorithm does not know the descent Mach and CAS values.

*TTF past first CAS*: The TTF is past the first procedurally CAS constrained waypoint on its route. This option is required to support the ATD-1 operational environment since it is assumed that the algorithm does not know the descent Mach and CAS values.

*Ownship past ABP*: The ownship is past the achieve-by point.

*CAS mode*: Since ASTAR does not have knowledge of the descent Mach and CAS values in the ATD-1 operational environment, the CTD control law is only valid in the CAS operational regime. The TBO portion of ASTAR determines if this condition has been met and provides the state value for this variable.

*Common path*: The history data from the TTF's state data and the ownship's current position relative to that data are used to determine if the ownship is on a common path with the TTF. The processing for the value of this variable is performed external to the control law selection process.

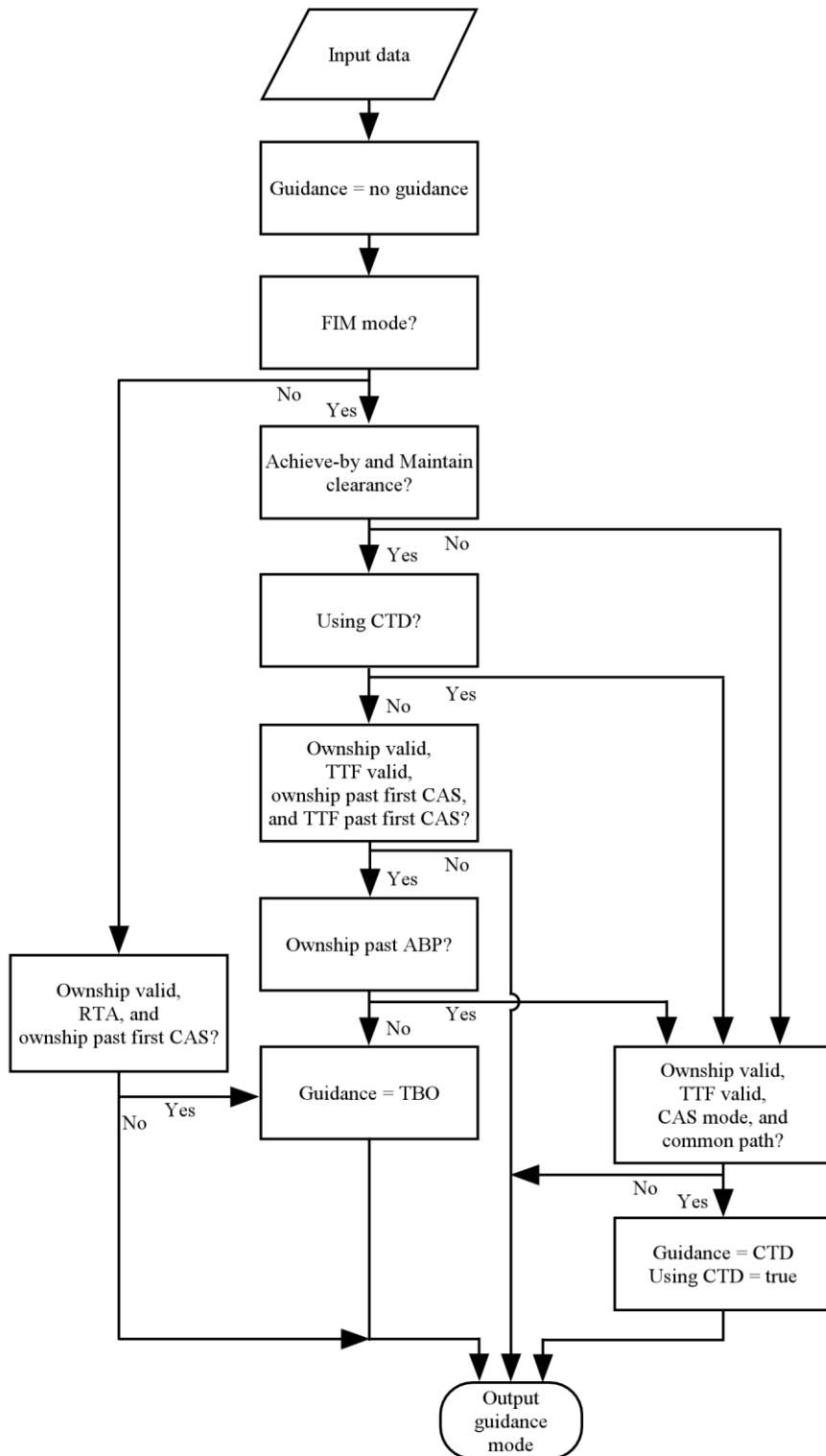


Figure 24. Control law selection diagram.

## Operational Considerations

### Common Speeds After Merging

The potential for the loss of separation or less than operationally desirable separation distances between the ownship and the TTF can be minimized by the design of the speed profiles on the respective 2D paths. The speeds specified in the path definitions at and after the point where the paths join in the horizontal plane must be the same speeds (fig. 25). That is, common path points must have common speeds.

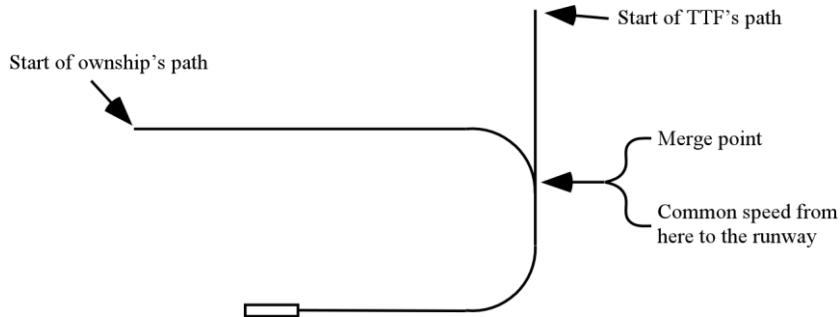


Figure 25. Example of common speeds after the merge point.

### Envelope Protection

Since the speed command value from ASTAR could be used to directly drive an autothrust system, speed envelope limiting can optionally be provided by the algorithm. To invoke this feature, the maximum and minimum desired speed values, both Mach and CAS, are input into ASTAR. These input limit speeds are usually based on the design limiting speed, the maximum gust penetration speed, the maximum flap extended speed, and minimum maneuvering speed. The algorithm then limits the command speed to remain within these values. When the command speed is limited, the algorithm sets an output flag indicating this limiting condition.

### Off-Nominal Mach / CAS Transition

During TBO, the algorithm provides both Mach and CAS speed command values and a Mach/CAS flag indicating which of these values, Mach or CAS, is appropriate for use relative to the aircraft's current flight conditions. While the 4D trajectory data provides the nominal altitude value for the Mach to CAS transition, this altitude value is only valid if the aircraft is exactly on the planned vertical path from the 4D trajectory and is at the nominal Mach. Because these conditions are not generally true, e.g., the Mach speed command is slower than the nominal value to correct a spacing error, ASTAR computes the Mach to CAS transition altitude for the current commanded speeds. Once the aircraft descends below this altitude, the algorithm transitions to a CAS command for the remainder of the operation.

## Future Design Considerations

Even with the design requirement to support a low cost, aircraft retrofit option with very minimal integration with other aircraft systems, several modifications could be made to ASTAR to reduce its design and implementation complexity along with supporting greater operational viability. Two of these modifications involve the calculations for the trajectory data and the third modification involves the speed command limiting, lag compensation, and output quantization.

### Estimated Time of Arrival Data from the Traffic to Follow

In the TBO portion of ASTAR, the assumption is that the TTF may only need minimal equipment, or possibly no extra equipment beyond ADS-B Out, to support airborne self-spacing. It was assumed that a normal self-spacing operation would start prior to the top of descent and continue through the start of the

ownship's deceleration to its final approach speed. To compute the trajectory data for the TTF, the ownship would need the following data for the TTF: the names describing its full path, i.e., arrival, transition, and approach names; its cruise Mach and altitude; and its planned final approach speed. It is also assumed that a database that is either local to or part of the ASTAR equipment would interpret the information describing the full path into data that are appropriate for the ASTAR trajectory computation. If these data required to calculate the TTF trajectory are not available via data link, other means of obtaining these data, or eliminating the need for these data, must be found.

Regardless of how these path data are obtained, the trajectory data calculated for the TTF is only a generic "guess" on how the TTF will actually fly the route. Discrepancies between the ASTAR trajectory calculation for the TTF and how the TTF actually flies the route, typically dictated by the FMS, would be propagated in ASTAR as a spacing error. One obvious option for eliminating a significant portion of the trajectory prediction error would be for the TTF to broadcast its ETA, or TTG, via data link. This broadcast could be done at a low frequency, e.g., once every 30 seconds, since the ETA value would typically not change very rapidly. This option would also eliminate the data requirements to the TTF's trajectory generation that were described in the previous paragraph. However, this option would require that the TTF have the ability to generate an accurate ETA and, obviously, the ability to data link this information.

### Trajectory Data from the Ownship FMS

Similar to the issues and benefits for using the TTF's ETA data from a data linked, FMS source, the use of the ownship's FMS data could significantly reduce the complexity of the spacing algorithm and increase its operational viability. The first obvious option in this regard would be to use the ownship's FMS ETA data in place of the ASTAR trajectory TTG. While this option may provide a more accurate TTG value, it still requires ASTAR to calculate a trajectory for the nominal speed values. As a second option, if the FMS could provide the ETA and the current, non-RTA nominal speed values, then ASTAR would not need to calculate a trajectory for the ownship. The most extensive option for reduction in ASTAR's complexity would come if the speed error value in figure 8 could either be used as an input to the FMS speed requirements, e.g., to adjust the next waypoint crossing speed, or superimposed on the FMS speed output command. Alternatively, the spacing time error value could be used to calculate a pseudo RTA value and this RTA value input into the FMS.

### Speed Prediction, Speed Quantization, and Lag Compensation Functions

Many of the ancillary functions in the TBO portion of ASTAR, e.g., the autothrust lag compensation, were added to the implementation ad hoc to overcome operational or performance issues that were observed prior to and during simulation evaluations. Several of these functions were designed to compensate for the lack of integration between the FIM algorithm and aircraft avionics in a retrofit implementation. As such, the overall design of these numerous functions were not designed "in the large." It would be beneficial for both the simplification of the algorithm and potentially better operational performance if these functions could be consolidated into one or two coherent functions.

### Summary

This paper provides an overview of the Airborne Spacing for Terminal Arrival Routes (ASTAR) algorithm. This algorithm is an airborne self-spacing tool that uses ADS-B data from a leading aircraft assigned by ATC to either achieve or maintain a precise spacing interval behind a self-spacing aircraft. The sixth revision of the ASTAR algorithm, ASTAR13, was tailored toward supporting the Air Traffic Management Technology Demonstration-1 (ATD-1) and to conform to the evolving industry Flight-deck Interval Management (FIM) standards document. This revised algorithm places significant emphasis on providing a low cost, retrofit avionics option that requires minimal integration with other aircraft systems. In this revision, a state-based speed control law was added to the original ASTAR trajectory-based control law to support new operational requirements from the evolving industry standards. Also, in support of the

ATD-1 concept it was assumed that the speed command value would be presented to the pilot and that the pilot would change the speed target of the autothrust system to match the commanded speed from ASTAR. Several capabilities are provided within the algorithm that attempt to balance the number of speed changes against the spacing performance. In addition to describing the trajectory computations, spacing interval calculations, and both the trajectory-based and state-based speed control laws, this paper discusses operational issues that were addressed in the development of the ASTAR algorithm.

## References

1. Sorensen, J. A., et al: Traffic Alert and Collision Avoidance System (TCAS) - Cockpit Display of Traffic Information (CDTI) Investigation, DOT/FAA/RD-91/8, April 1991.
2. Swedish, W. J.: *Evaluation of the Potential for Reduced Longitudinal Spacing on Final Approach*, FAA-EM-79-7, August 1979.
3. Abbott, Terence S.: Speed Control Law for Precision Terminal Area In-Trail Self-Spacing. NASA/TM-2002-211742, 2002.
4. Oseguera-Lohr, Rosa M.; Lohr, Gary W.; Abbott, Terence S.; Nadler, Eric D.; and Eischeid, Todd M.: Evaluation of a Tool for Airborne-Managed In-Trail Approach Spacing. NASA/TP-2005-213773, August 2005.
5. Barmore, Bryan E.; Abbott, Terence S.; Capron, William R.: Evaluation of Airborne Precision Spacing in a Human-in-the-Loop Experiment. NASA Center for AeroSpace Information (CASI), AIAA Paper 2005-7402; AIAA 5th Aviation Technology, Integration, and Operations Conference (ATIO), Arlington, VA, United States, 26-28 Sep. 2005.
6. Abbott, Terence S.: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing to Include Parallel Runway Operations. NASA/CR-2011-217194, 2011.
7. *Air Traffic Control*. DOT/FAA Order JO 7110.65T, Feb. 11, 2010.
8. Abbott, Terence S.: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Third Revision. NASA/CR-2012-217786, 2012.
9. Abbott, Terence S.: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Fourth Revision. NASA/CR- 2013-218044, 2013.
10. Baxley, B.T.; Johnson, . C.; Swenson, H. N.; Robinson, J. E.; Prevot, T.; Callantine, T. J.; Scardina, J.; and Greene, M.: Air Traffic Management Technology Demonstration-1 Concept of Operations (ATD-1 ConOps). NASA/TM-2012-217585, 2012.
11. Abbott, Terence S: An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Fifth Revision. NASA/ CR-2015-218680, February 2015.
12. Federal Aviation Administration, NextGen Mid-Term Concept of Operations for the National Airspace System, FAA ATO, Version 2.0, April 30, 2010.
13. Abbott, Terence S: A Brief History of Airborne Self-Spacing Concepts, NASA/CR-2009-215695, 2009.
14. CoSpace 2003 Aircraft Guidance Model Based Experiments Evaluation of Airborne Spacing For Traffic Sequencing, EEC Report No. 391, 2004.
15. Kelly, James R. and Abbott, Terence S.: In-Trail Spacing Dynamics of Multiple CDTI-Equipped Aircraft Queues, NASA TM-85699, 1984.
16. Williams, David H. and Well, Douglas C.: Jet Transport Flight Operations Using Cockpit Display of Traffic Information During Instruments Meteorological Conditions, Simulation Evaluation, NASA TP-2567, 1986.
17. RTCA: Minimum Operational Performance Standards (MOPS) for Flight-deck Interval Management (FIM), pre-FRAC draft, 2014.
18. Williams, David H.: Time-Based Self-Spacing Techniques Using Cockpit Display of Traffic Information During Approach to Landing in a Terminal Area Vectoring Environment, NASA TM 84601, 1983.
19. *Safety, Performance and Interoperability Requirements Document for Airborne Spacing – Flight Deck Interval Management (ASPA-FIM)*. RTCA DO-328, 2011.
20. Abbott, Terence S.: A Trajectory Algorithm to Support En Route and Terminal Area Self-Spacing Concepts: Third Revision. NASA/CR- 2014-2182884, 2014.
21. *Instrument Procedures Handbook*. DOT/FAA-H-8261-1A, 2007.

22. Norris, Guy: *UPS to Test 'SafeRoute' Next Year*, Flight International Magazine, p. 12, 9-15 Aug. 2005.
23. Swieringa, Kurt; Murdoch, Jennifer L.; Baxley, Brian; and Hubbs, Clay: Evaluation of an Airborne Spacing Concept, On-Board Spacing Tool, and Pilot Interface. 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Virginia Beach, VA, 20-22 Sep. 2011.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE		3. DATES COVERED (From - To)		
01-07 - 2015	Contractor Report				
4. TITLE AND SUBTITLE			5a. CONTRACT NUMBER NNL10AA14B 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER 5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER 305295.02.41.07.03		
An Overview of a Trajectory-Based Solution for En Route and Terminal Area Self-Spacing: Sixth Revision					
6. AUTHOR(S)			5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER 305295.02.41.07.03		
Abbott, Terence S.					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)			8. PERFORMING ORGANIZATION REPORT NUMBER		
NASA Langley Research Center Hampton, Virginia 23681-2199					
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S) NASA 11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/CR-2015-218774		
National Aeronautics and Space Administration Washington, DC 20546-0001					
12. DISTRIBUTION/AVAILABILITY STATEMENT					
Unclassified Subject Category 03 Availability: NASA STI Program (757) 864-9658					
13. SUPPLEMENTARY NOTES					
Langley Technical Monitor: Kurt A. Swieringa					
14. ABSTRACT					
<p>This paper presents an overview of the sixth revision to an algorithm specifically designed to support NASA's Airborne Precision Spacing concept. This algorithm is referred to as the Airborne Spacing for Terminal Arrival Routes version 13 (ASTAR13). This airborne self-spacing concept contains both trajectory-based and state-based mechanisms for calculating the speeds required to achieve or maintain a precise spacing interval. The trajectory-based capability allows for spacing operations prior to the aircraft being on a common path. This algorithm was also designed specifically to support a standalone, non-integrated implementation in the spacing aircraft. This current revision to the algorithm adds the state-based capability in support of evolving industry standards relating to airborne self-spacing.</p>					
15. SUBJECT TERMS					
Aircraft operations; Aircraft systems; Approach spacing					
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON STI Help Desk (email: <a href="mailto:help@sti.nasa.gov">help@sti.nasa.gov</a> ) 19b. TELEPHONE NUMBER (Include area code) (757) 864-9658	
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	50	
U	U	U			